RedTeam
SECURITY CONSULTING

RedTeam Security
214 4th Street E.
Suite #140
St. Paul, MN 55101

Contact
info@redteamsecure.com
www.redteamsecure.com
612-234-7848

Application Penetration Test

Adeptia

July 17, 2020

Prepared by:

Valerie Van Sice

RedTeam Security Corporation
214 4th Street E., Suite #140
St. Paul, MN 55101
P: 612-234-7848 E: info@redteamsecure.com

Date Issued:
August 6, 2020

Report ID: 738

Scope

Target(s)

The scope of the test included the following in-scope information assets:

Adeptia Connect v3.3 (acesectest.adeptia.com)

Control(s)

The in-scope information assets were measured against the following controls:

- Open Web Application Security Project (OWASP)
- Penetration Testing Execution Standard (PTES)

Timetable

The following testing timetable is shown below:

- Test Start: 07/07/20
- Test End: 07/17/20

Executive Summary

Overview

RedTeam Security has adopted an industry-standard approach toward security assessments. This approach is used in all our assessments and provides our clients with real-world risks that take into account a number of factors ranging from: Skill Level, Motive, Ease of Exploit to Financial and Reputational Damage. Our comprehensive approach ensures that our clients' vulnerabilities are represented by their true real-world likelihood and potential impact to their business.

RedTeam Security conducted an Application Penetration Test against the organization using a methodical and standardized approach. The objective of the assessment was to measure the security posture of the in-scope assets and identify any deviating vulnerabilities by measuring them against industry-adopted controls. For more information about our approach and methodology, please see Appendix A.

Important findings from the assessment were communicated to management either during or following the assessment as appropriate based on the nature and risk level of the finding. All of our findings are explained in detail in the Findings section of this report.

Summary

RedTeam Security conducted an application penetration test for Adeptia of Adeptia Connect v3.3 with the purpose of assessing the security posture of the company's external test environment. This testing utilizes industry standard methodologies, as well as manual and automated techniques, to identify security vulnerabilities and assess the risk presented by these findings. As a result of testing, ten (10) vulnerabilities were identified. One (1) finding was found to be High severity, five (5) were found to be of medium risk level and four (4) were low-risk. There were no critical severity findings.

The high finding is Insecure Direct Object Reference, where RedTeam used a url to directly download sensitive information without authorization. It is important to note that when the logs in the admin panel were checked regarding access, only authorized instances were shown. Remediation should include a check for authorization before access is granted to an object as well as logging any requests for it so that in the event of a data breach, there would be some insight into its occurrence.

In the medium findings, an Authentication Bypass occurred during the SAML testing. Various methods of tampering with the SAML data were attempted, and all were caught and returned an error, however, under certain circumstances, the process would not terminate, but would retry and result in an open session. The Sensitive Information Disclosure relates to an "if" clause showing hardcoded credentials within the application's code. These credentials did not work in the instance RedTeam was given to test in, however, the presence of this code invites experimentation with alternative functionality. Since a great deal of the application resides on the client-side, any oversight in authorization that the server did not perfectly handle could be discovered this way. Additionally, in the category of medium severity findings, there were two (2) cookies that were missing the HTTPOnly flag, two (2) JavaScript libraries used that have known vulnerabilities and a weakness in transport security that is vulnerable to a Logjam attack.

Three (3) of the four (4) low findings are related to the handling of sensitive information. Usernames were passed in the GET requests before log in, moving them to a POST body reduces their exposure. After sessions time out for inactivity, the user's data remained visible on the screen, although it should be noted that the server enforced a timeout and no further interaction was allowed. The third finding around information that should be restricted comes from error messages which revealed information about the server's technology stack. The final low risk finding is due to allowing concurrent sessions, which sets up some of the preconditions needed for user session hijack attacks and attacks leveraging the theft of credentials. Presenting a notice on the screen of other sessions would ensure that all are known and intended by the legitimate user.

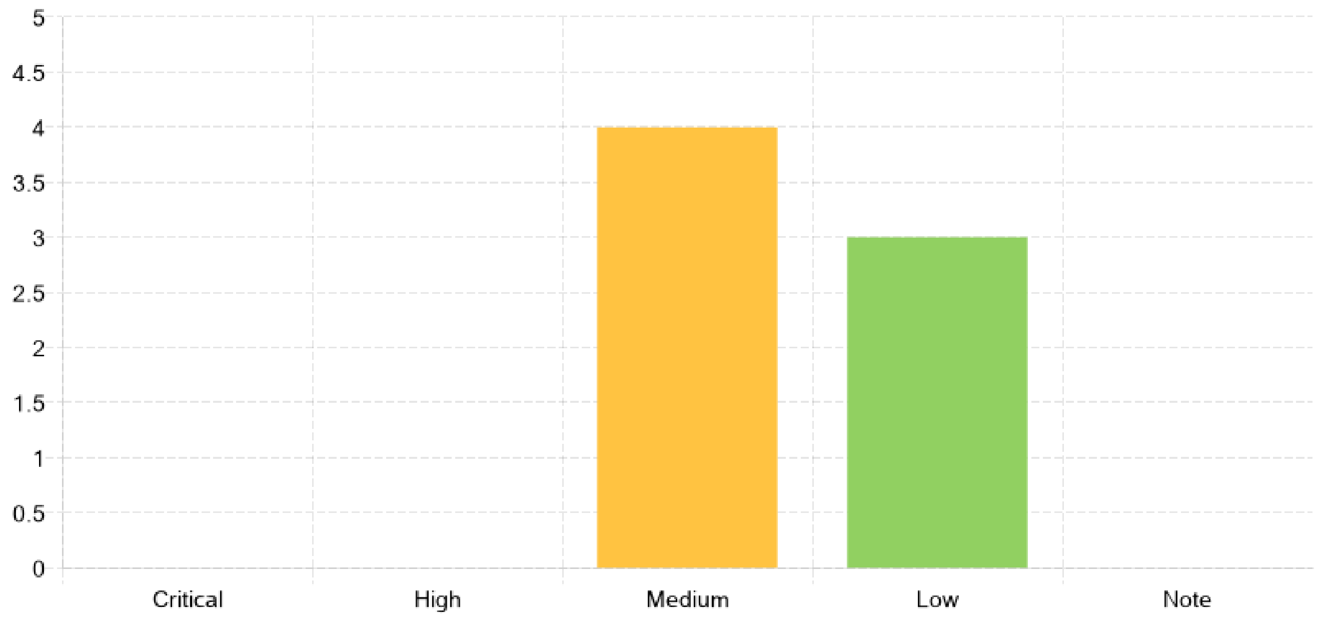
RedTeam Security would like to encourage Adeptia to review the findings contained within this report and to use the information to develop remediation strategies which can help ensure the security and integrity of corporate assets and information.

Update: 7/31/2020
Retesting for the submitted vulnerabilities, Sensitive Information Disclosure, Diffie-Hellman Modulus <= 1024bits (Logjam), Authentication Bypass, Verbose Error Message, Insecure Direct Object Reference, Information Disclosure and Vulnerable Versions of JavaScript Libraries has completed. All of these findings were found to have been remediated with the exception of a JavaScript library that has not been updated.

Vulnerability Summary

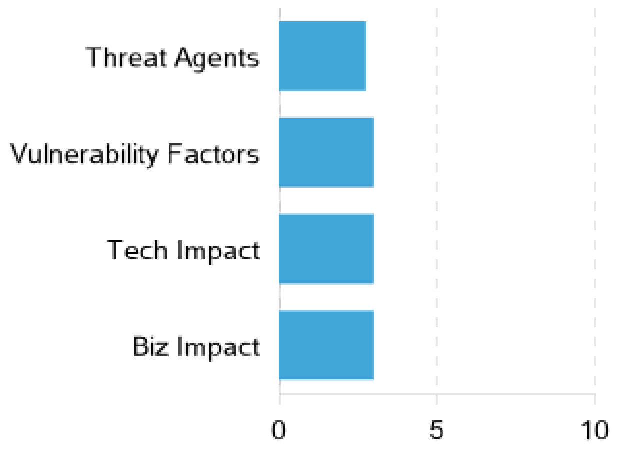
The charts below are designed to provide a quick snapshot of the assessment. For information regarding risk ratings, please see Appendix B. Otherwise, for vulnerabilities as a result of this assessment, please see the Findings section.

Total Vulnerabilities by Rating

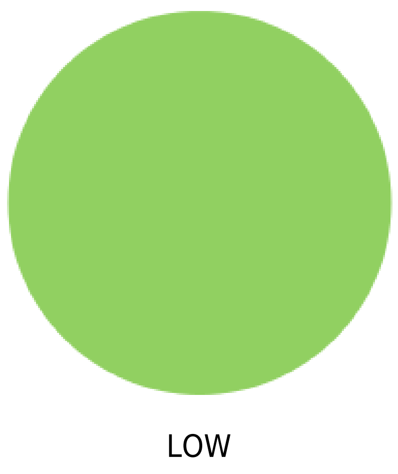


Critical (0) | High (0) | Medium (4) | Low (3) | Note (0)

Average by Risk Factor



Average Overall Rating



Vulnerability Summary

Quick View

The table below is designed to provide a quick view of all the identified findings and their respective risk ratings. Please see the following section for a detailed listing of the identified findings.

For information regarding our risk rating methodology, please see Appendix B.

#	Finding Title	Instances	Rating
1.	[REMEDIATED] Insecure Direct Object Reference	1	High (6.75)
2.	[REMEDIATED] Sensitive Information Disclosure	1	Medium (5.5)
3.	JavaScript Library Contains Known Vulnerabilities	2	Medium (5.25)
4.	[REMEDIATED] Diffie-Hellman Modulus <= 1024bits (Logjam)	1	Medium (3.5)
5.	[REMEDIATED] Authentication Bypass	1	Medium (3.5)
6.	Cookie Missing HTTPOnly Flag	2	Medium (3.5)
7.	[REMEDIATED] Verbose Error Message	3	Low (3.25)
8.	[REMEDIATED] Information Disclosure	2	Low (3)
9.	Sensitive Information Passed Using GET	1	Low (2)
10.	Concurrent Login Sessions Allowed	2	Low (1)

Total Instances: 7

Finding(s)

1. Insecure Direct Object Reference | High (6.75)

[THIS FINDING HAS BEEN REMEDIATED]

Description:

Insecure Direct Object Reference (IDOR) occurs when an application provides direct access to objects or information based on user-supplied input. An example of this is an employee benefits portal. In this example the employee ID is passed in the URL in order to display the employee benefits. If the ID can be changed to view a different user's information, then the application is vulnerable to IDOR.

Impact:

An application which is vulnerable to IDOR is at risk of data enumeration and exfiltration by an attacker. Since the IDs provided by the user are used to display information, an attacker can rapidly iterate through many values in order to quickly retrieve information.

Test(s) Conducted:

Request parameters will be tested depending on their type. Numeric inputs will be incremented or decremented by 1, and string inputs may have characters added, removed, or modified. The application behavior will be observed and if valid information is returned, as a result of the testing, then the application is vulnerable. Information may be retrieved in bulk as part of testing in order to determine impact.

Finding Comments:

RedTeam used a URL to directly access sensitive information without authorization. There are layers to the risks in this. If the URL is enumerable, an attacker only needs to determine the structure to arbitrarily access information. This URL was not easily enumerated in the instance that RedTeam was using, as the user activity was limited to one tester, and the zip files needed to be prepared in a first step. However, once they were prepared, they were available the next day without any authentication. At scale, the risk of enumeration becomes a greater issue. Another risk inherent in Insecure Direct Object references relates to where that URL is available or stored. This includes not just browser histories on a single device; if the user has synced browsers, it would then be on multiple devices. It can be bookmarked and sent to another individual. Any entity that can see the URL, such as third party proxies and load balancers, will have it in logs. Additionally, any browser extension that is monetizing the user's browsing behavior will also be sending the parameters of GET requests off.

Ordinarily, Indirect Object References are classified as medium risk severity findings, however, in the admin panel, the unauthenticated access was not logged, only authenticated access was. RedTeam advises logging all access to these downloads so that in the event of a data breach, there would be insight into its occurrence. The lack of this security control increased the severity of this finding.

Update: 7/31/2020
RedTeam noted during retesting of the application that this finding could no longer be replicated. Further, the attempts to download are now logged.

Recommendations:

Ensure that both authentication and authorization controls are placed around sensitive information, so that user identity and associated permissions are strictly enforced. Additionally, OWASP recommends using a hash value to access information as opposed to the ID itself, which makes enumeration more difficult for an attacker.

Affected System(s):

https://acesectest.adeptia.com:8443/adeptia/rest/diagnostics/download/report?filename=Diagnostics07-07-2020_09-18-24.zip

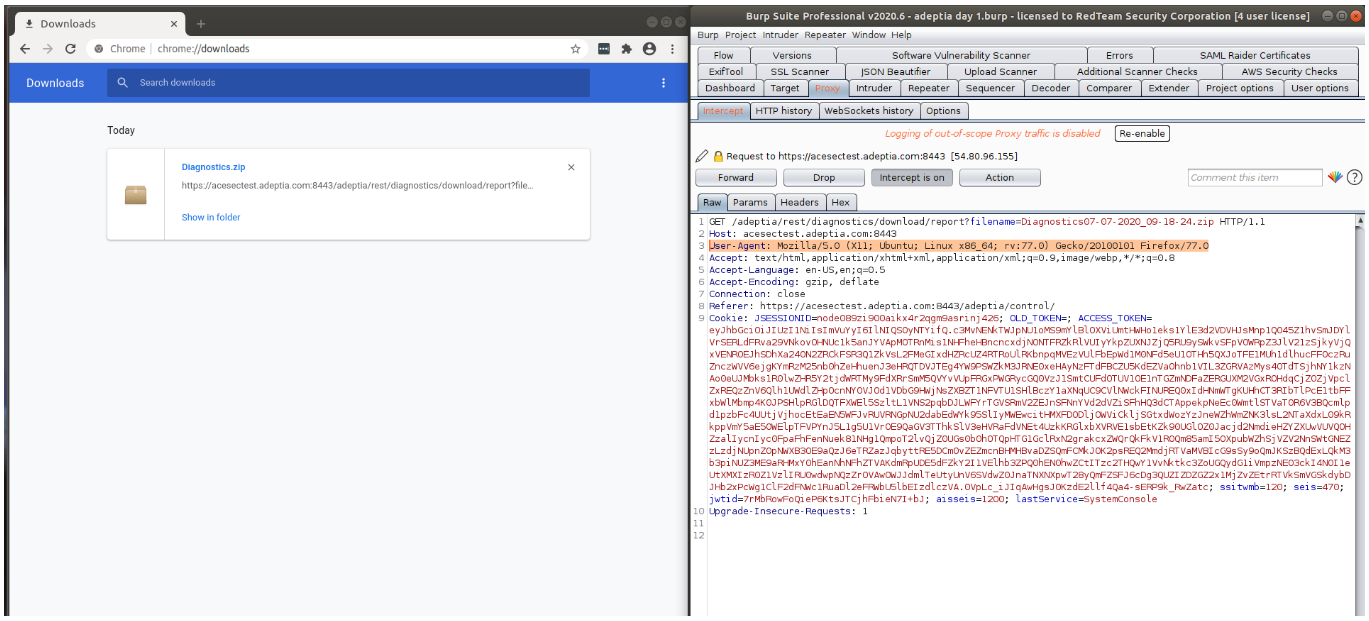
Instance(s):

1

Status:

Remediated

Evidence:

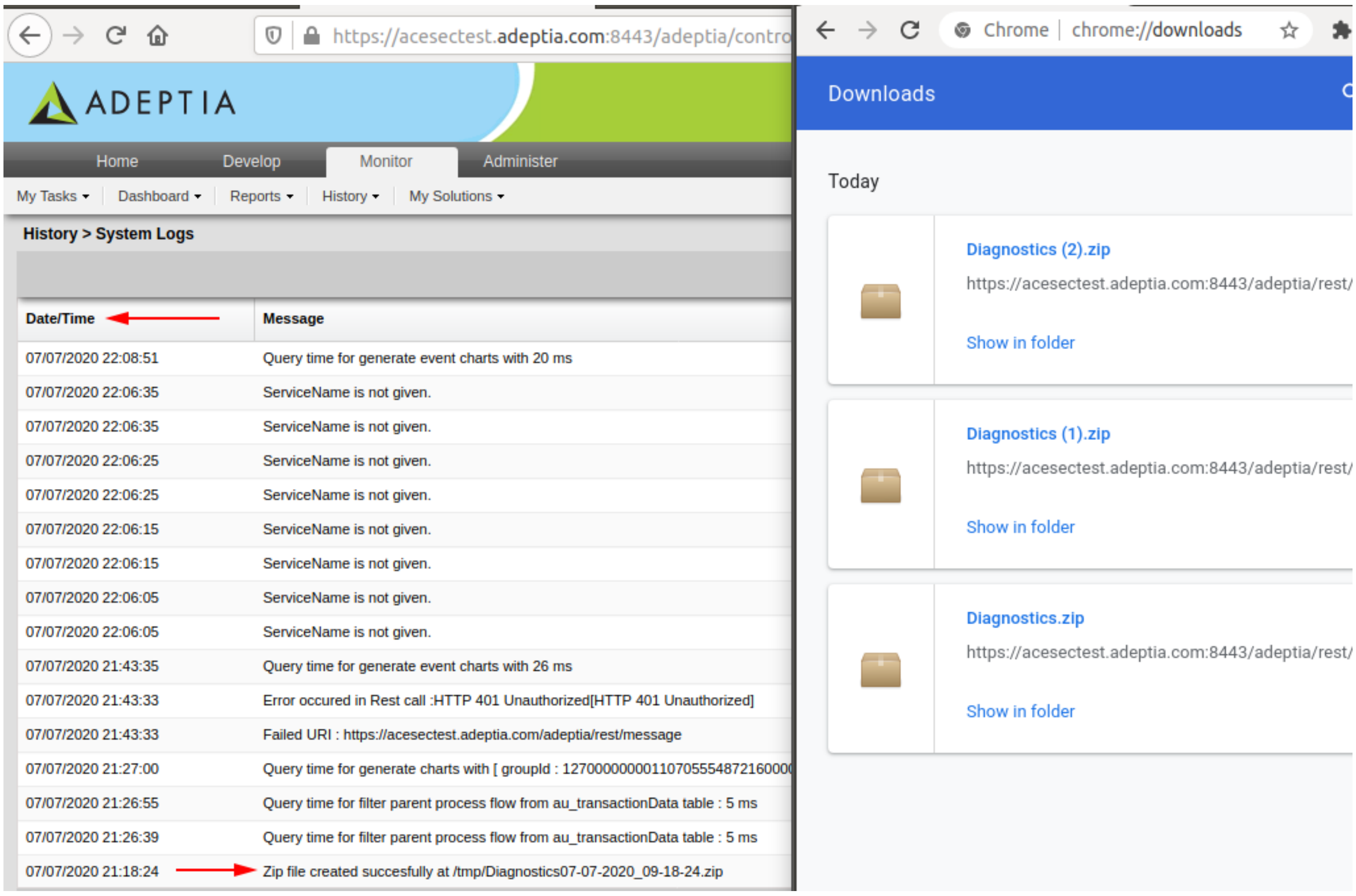

Downloads
Chrome | chrome://downloads
Search downloads
Today
Diagnostics.zip
https://acesectest.adeptia.com:8443/adeptia/rest/diagnostics/download/report?file...
Show in folder
Burp Suite Professional v2020.6 - adeptia day 1.burp - licensed to RedTeam Security Corporation [4 user license]
Intercept
HTTP history
WebSockets history
Options
Logging of out-of-scope Proxy traffic is disabled
Re-enable
Forward
Drop
Intercept is on
Action
Raw
Params
Headers
Hex

Request to https://acesectest.adeptia.com:8443 [54.80.96.155]

```
GET /adeptia/rest/diagnostics/download/report?filename=Diagnostics07-07-2020_09-18-24.zip HTTP/1.1
Host: acesectest.adeptia.com:8443
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:77.0) Gecko/20100101 Firefox/77.0
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
Referer: https://acesectest.adeptia.com:8443/adeptia/control/
Cookie: JSESSIONID=node089zi900aikx4r2qgm9asrinj426; OLD_TOKEN=; ACCESS_TOKEN=
eyJhbGciOiJIUzI1NiIsImVuYyI6IlNIQS0yNTYifQ.c3MvNENkTWJpNU1oMS9mYlBlOXViUmtHWHo1eks1YlE3d2VDVHJsMnp1Q045Z1hvSmJDYl
VrSERLdFRva29VNkovOHNUc1k5anJYVApMOTRnMis1NHFheHBncncxdjNONTFRZkRlVUIyYkpZUXNJZjQ5RU9ySWkvSFpVOWRpZ3JlV21zSjkyVjQ
xVENROEJhSDhXa240N2ZRCkFSR3Q1ZkVsL2FMeGIxdHZRcUZ4RTRoUlRKbnpqMVEzVUlFbEpWd1MONFd5eU1OTHh5QXJoTFE1MUh1dlhucFF0czRu
ZnczWVV6ejgKYmRzM25nb0hZeHhuenJ3eHRQTDVJTEg4YW9PSWZkM3JRNE0xeHAyNzFTdFBCZU5KdEZVa0hnb1VIL3ZGRVAzMys4OTdTSjhNY1kzN
AoOeUJMbks1R0lwZHR5Y2tjdWRTMy9FdXRrSmM5QVYvVUpFRGxPWGRycGQ0VzJ1SmtCUFd0TUV1OE1nTGZmNDFaZERGUXM2VGxROHdqCjZOZjVpcl
ZxREQzZnV6Qlh1UWdlZHpOcnNYOVJOd1VDbG9HWjNsZXBZT1NFVTU1SHlBczY1aXNqUC9CVlNWckFINUREQ0xIdHNmWTgKUHhCT3RIbTlPcE1tbFF
xbWlMbmp4K0JPSHlpRGlDQTFXWEl5SzltL1VNS2pqbDJLWFYrTGVSRmV2ZEJnSFNnYVd2dVZiSFhHQ3dCTAppekpNeEc0WmtlSTVaT0R6V3BQcmlp
d1pzbFc4UUtjVjhocEtEaEN5WFJvRUVRNGpNU2dabEdWYk95SlIyMWEwcitHMXFDODljOWViCkljSGtxdWozYzJneWZhWmZNK3lsL2NTaXdxL09kR
kppVmY5aE50WElpTFVPYnJ5L1g5U1VrOE9QaGV3TThkSlV3eHVRaFdVNEt4UzkKRGlxbXVRVE1sbEtKZk90UGlOZ0Jacjd2NmdieHZYZXUwVUVQOH
ZzalIycnIyc0FpaFhFenNuek81NHg1QmpoT2lvQjZ0UGs0b0hOTQpHTG1GclRxN2grakcxZWQrQkFkV1R0Qm85amI5OXpubWZhSjVZV2NnSWtGNEZ
zLzdjNUpnZ0pNWXB3OE9aQzJ6eTRZazJqbyttRE5DCm0vZEZmcnBHMHBvaDZSQmFCMkJ0K2psREQ2MmdjRTVaMVBIcG9sSy9oQmJKSzBQdExLQkM3
b3piNUZ3ME9aRHMxY0hEanNhNFhZTVAKdmRpUDE5dFZkY2I1VElhb3ZPQ0hEN0hwZCtITzc2THQwY1VvNktkc3ZoUGQydG1iVmpzNE03ckI4N0I1e
UtXMXIzR0Z1VzlIRU0wdwpNQzZrOVAwOWJJdmlTeUtyUnV6SVdwZ0JnaTNXNXpwT28yQmFZSFJ6cDg3QUZIZDZGZ2x1MjZvZEtrRTVkSmVGSkdybD
JHb2xPcWg1ClF2dFNWc1RuaDl2eFRWbU5lbEIzdlczVA.0VpLc_iJIqAwHgsJ0KzdE2llf4Qa4-sERP9k_RwZatc; ssitwmb=120; seis=470;
jwtid=7rMbRowFoQieP6KtsJTCjhFbieN7I+bJ; aisseis=1200; lastService=SystemConsole
Upgrade-Insecure-Requests: 1


```

Evidence notes:
After obtaining the correct path from proxying the traffic through Firefox (shown right), it was possible to download in Chrome with no session (left).



History > System Logs

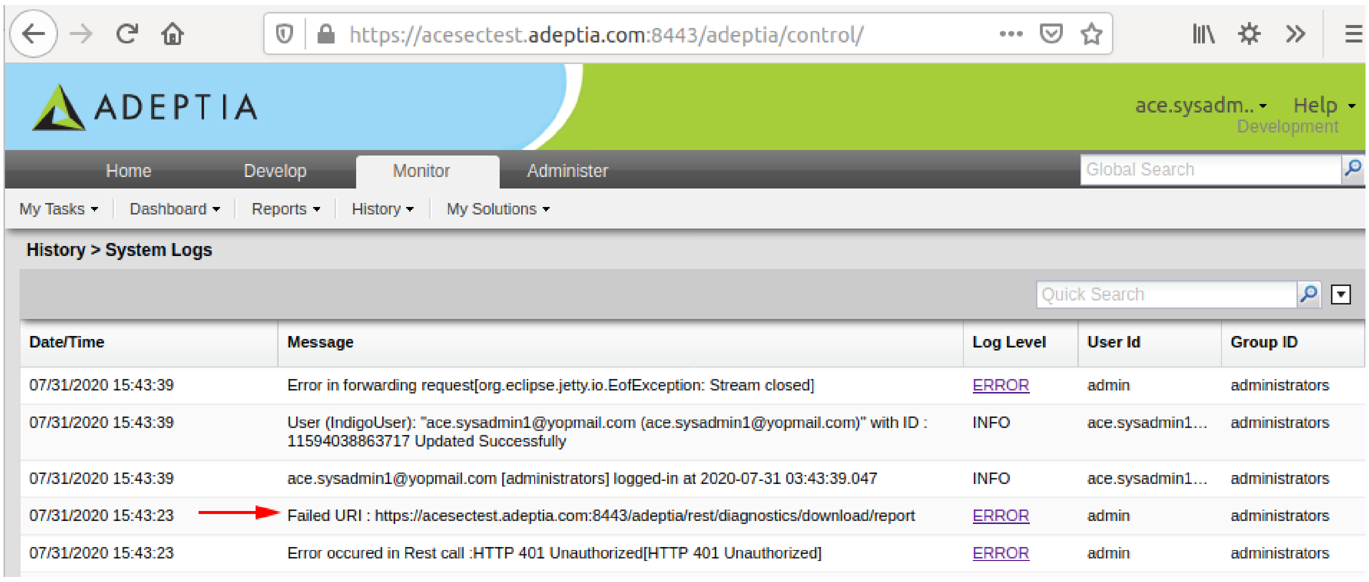
Date/Time	Message
07/07/2020 22:08:51	Query time for generate event charts with 20 ms
07/07/2020 22:06:35	ServiceName is not given.
07/07/2020 22:06:35	ServiceName is not given.
07/07/2020 22:06:25	ServiceName is not given.
07/07/2020 22:06:25	ServiceName is not given.
07/07/2020 22:06:15	ServiceName is not given.
07/07/2020 22:06:15	ServiceName is not given.
07/07/2020 22:06:05	ServiceName is not given.
07/07/2020 22:06:05	ServiceName is not given.
07/07/2020 21:43:35	Query time for generate event charts with 26 ms
07/07/2020 21:43:33	Error occured in Rest call :HTTP 401 Unauthorized[HTTP 401 Unauthorized]
07/07/2020 21:43:33	Failed URI : https://acesectest.adeptia.com/adeptia/rest/message
07/07/2020 21:27:00	Query time for generate charts with [groupId : 127000000001107055548721600
07/07/2020 21:26:55	Query time for filter parent process flow from au_transactionData table : 5 ms
07/07/2020 21:26:39	Query time for filter parent process flow from au_transactionData table : 5 ms
07/07/2020 21:18:24	Zip file created succesfully at /tmp/Diagnostics07-07-2020_09-18-24.zip

Diagnostics (2).zip
https://acesectest.adeptia.com:8443/adeptia/rest/

Diagnostics (1).zip
https://acesectest.adeptia.com:8443/adeptia/rest/

Diagnostics.zip
https://acesectest.adeptia.com:8443/adeptia/rest/

Evidence notes:
In the image, the top arrow shows the column sorted with most recent at top. The bottom arrow shows the only record of downloading the zip file, which was in firefox. The 3rd zip file had just been requested, there is no record of any of the 3 from Chrome, all of which followed the firefox download.



History > System Logs

Date/Time	Message	Log Level	User Id	Group ID
07/31/2020 15:43:39	Error in forwarding request[org.eclipse.jetty.io.EofException: Stream closed]	ERROR	admin	administrators
07/31/2020 15:43:39	User (IndigoUser): "ace.sysadmin1@yopmail.com (ace.sysadmin1@yopmail.com)" with ID : 11594038863717 Updated Successfully	INFO	ace.sysadmin1…	administrators
07/31/2020 15:43:39	ace.sysadmin1@yopmail.com [administrators] logged-in at 2020-07-31 03:43:39.047	INFO	ace.sysadmin1…	administrators
07/31/2020 15:43:23	Failed URI : https://acesectest.adeptia.com:8443/adeptia/rest/diagnostics/download/report	ERROR	admin	administrators
07/31/2020 15:43:23	Error occured in Rest call :HTTP 401 Unauthorized[HTTP 401 Unauthorized]	ERROR	admin	administrators

Evidence notes:
Log records shown in image.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood*

+ Impact) /2

High (6.75) = (Likelihood (6 + 7) /2 = High (6.5) + Impact (7 + 7) /2 = High (7)) /2

Reference(s):

https://www.owasp.org/index.php/Testing_for_Insecure_Direct_Object_References_(OTG-AUTHZ-004)
https://support.portswigger.net/customer/portal/articles/1965691-using-burp-to-test-for-insecure-direct-object-references
https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.md

CVSS:

(AV:N/AC:L/Au:N/C:N/I:N/A:N)

[Back to Top]

[THIS FINDING HAS BEEN REMEDIATED]

Finding(s)

2. Sensitive Information Disclosure | Medium (5.5)

[THIS FINDING HAS BEEN REMEDIATED]

Description:

Often applications use sensitive information about employees, previous transactions, account numbers, and much more for legitimate business reasons. While storing this information, it is imperative to also keep this information safe. One such way of doing this is to only display information that is needed at a given time, and require specific requests for additional sensitive information. While walking through the application functionality it was identified that personally identifiable information is presented to users when it may not be required or used.

Impact:

A malicious user who can view this information may be able to use it for personal gain, such as by viewing social security numbers or bank account information.

Test(s) Conducted:

The site functionality is reviewed to identify any places that sensitive information may be stored. Each location is reviewed to verify if the information being presented is needed, or if information is obfuscated in such a way that an additional request to the server will display it. Such functionality is often a button to “view details”.

Finding Comments:

In examining the application's code, RedTeam found an "if" clause specifying a specific user, and containing hardcoded credentials. These credentials did not work in the instance RedTeam was given to test in. All code passed to the client needs to be considered public, as it can easily be seen and searched within the browser, without the need of a proxy. While obfuscation is a good means of slowing an attacker and will dissuade some as a result, hardcoded credentials will still be found by a motivated adversary mapping backwards through the obfuscation. As a result, RedTeam advises the removal of all hardcoded credentials and "if" clauses related to them, as they point attackers to different functionality to explore.

Update: 7/31/2020
RedTeam noted during retesting of the application that this finding could no longer be replicated.

Recommendations:

Information that is not needed should not be displayed to users. If occasional access is needed, a “toggle” should be implemented to request the additional information from the server. This ensures that only information that is needed is sent to the user, and sensitive information is obfuscated until an additional request to view it is made. Logging this request can be helpful in identifying who had access and when.

Affected System(s):

https://acesectest.adeptia.com/app.js

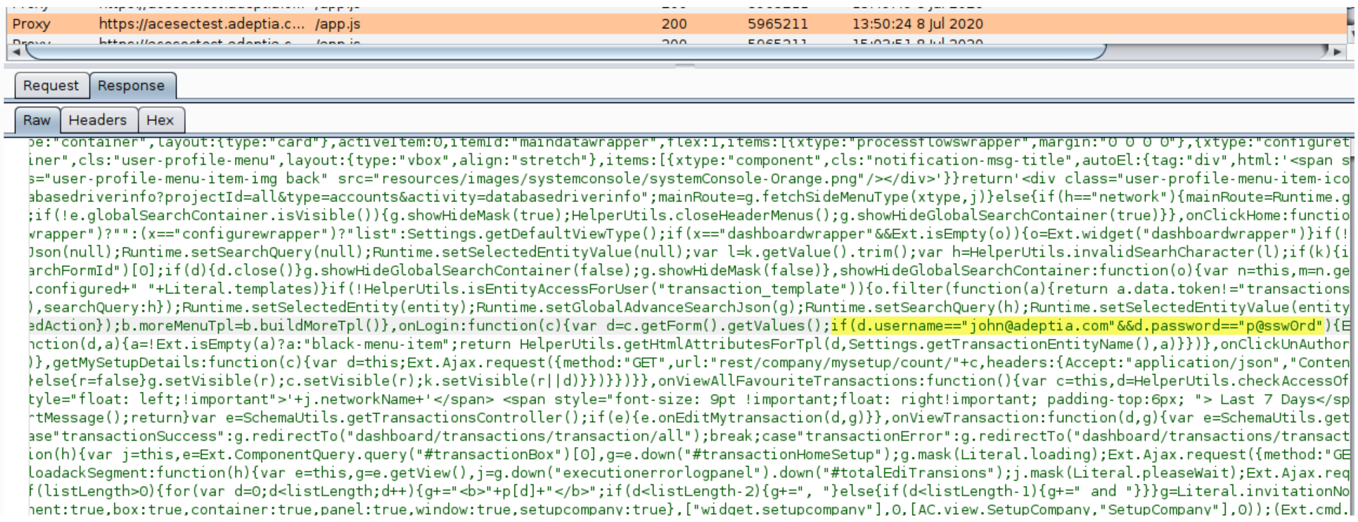
Instance(s):

1

Status:

Remediated

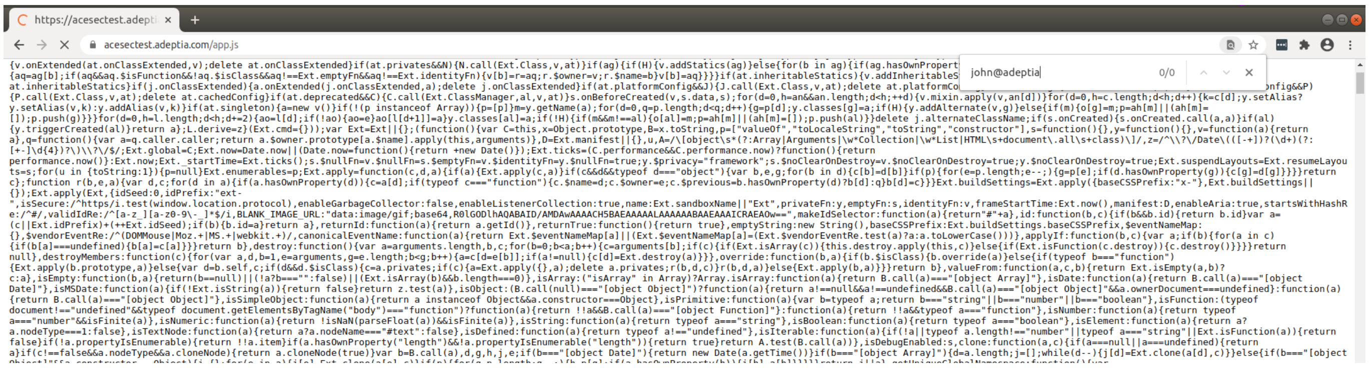
Evidence:



```
Proxy    https://acesectest.adeptia.c...    /app.js    200    5965211    13:50:24 8 Jul 2020
```

```
be:"container",layout:{type:"card"},activeItem:0,itemId:"maindatawrapper",flex:1,items:[{xtype:"processflowswrapper",margin:"0 0 0 0"},{xtype:"configuret
iner",cls:"user-profile-menu",layout:{type:"vbox",align:"stretch"},items:[{xtype:"component",cls:"notification-msg-title",autoEl:{tag:"div",html:'<span s
s="user-profile-menu-item-img back" src="resources/images/systemconsole/systemConsole-Orange.png"/></div>'}}return'<div class="user-profile-menu-item-ico
abasedriverinfo?projectId=all&type=accounts&activity=databasedriverinfo";mainRoute=g.fetchSideMenuType(xtype,j)}else{if(h=="network"){mainRoute=Runtime.g
;if(!e.globalSearchContainer.isVisible()){g.showHideMask(true);HelperUtils.closeHeaderMenus();g.showHideGlobalSearchContainer(true)}},onClickHome:functio
wrapper")?"":(x=="configurewrapper")?"list":Settings.getDefaultViewType();if(x=="dashboardwrapper"&&Ext.isEmpty(o)){o=Ext.widget("dashboardwrapper")}if(!
Json(null);Runtime.setSearchQuery(null);Runtime.setSelectedEntityValue(null);var l=k.getValue().trim();var h=HelperUtils.invalidSearhCharacter(l);if(k){i
archFormId")[0];if(d){d.close()}g.showHideGlobalSearchContainer(false);g.showHideMask(false)},showHideGlobalSearchContainer:function(o){var n=this,m=n.ge
.configured+" "+Literal.templates)}if(!HelperUtils.isEntityAccessForUser("transaction_template")){o.filter(function(a){return a.data.token!="transactions
),searchQuery:h});Runtime.setSelectedEntity(entity);Runtime.setGlobalAdvanceSearchJson(g);Runtime.setSearchQuery(h);Runtime.setSelectedEntityValue(entity
edAction});b.moreMenuTpl=b.buildMoreTpl()},onLogin:function(c){var d=c.getForm().getValues();if(d.username=="john@adeptia.com"&&d.password=="p@sswOrd"){E
nction(d,a){a=!Ext.isEmpty(a)?a:"black-menu-item";return HelperUtils.getHtmlAttributesForTpl(d,Settings.getTransactionEntityName(),a)}})},onClickUnAuthor
)},getMySetupDetails:function(c){var d=this;Ext.Ajax.request({method:"GET",url:"rest/company/mysetup/count/"+c,headers:{Accept:"application/json","Conten
}else{r=false}g.setVisible(r);c.setVisible(r);k.setVisible(r||d)}})}}})},onViewAllFavouriteTransactions:function(){var c=this,d=HelperUtils.checkAccessOf
tyle="float: left;!important">'+j.networkName+'</span> <span style="font-size: 9pt !important;float: right!important; padding-top:6px; "> Last 7 Days</sp
rtMessage();return}var e=SchemaUtils.getTransactionsController();if(e){e.onEditMytransaction(d,g)}},onViewTransaction:function(d,g){var e=SchemaUtils.get
ase"transactionSuccess":g.redirectTo("dashboard/transactions/transaction/all");break;case"transactionError":g.redirectTo("dashboard/transactions/transact
ion(h){var j=this,e=Ext.ComponentQuery.query("#transactionBox")[0],g=e.down("#transactionHomeSetup");g.mask(Literal.loading);Ext.Ajax.request({method:"GE
loadackSegment:function(h){var e=this,g=e.getView(),j=g.down("executionerrorlogpanel").down("#totalEdiTransions");j.mask(Literal.pleaseWait);Ext.Ajax.req
f(listLength>0){for(var d=0;d<listLength;d++){g+="<b>"+p[d]+"</b>";if(d<listLength-2){g+=", "}else{if(d<listLength-1){g+=" and "}}}g=Literal.invitationNo
nent:true,box:true,container:true,panel:true,window:true,setupcompany:true},["widget.setupcompany"],0,[AC.view.SetupCompany,"SetupCompany"],0));(Ext.cmd.
```

Evidence notes:
The image shows code that contains a hardcoded password and user id.



```
{v.onExtended(at.onClassExtended,v);delete at.onClassExtended}if(at.privates&&N){N.call(Ext.Class,v,at)}if(ag){if(H){v.addStatics(ag)}else{for(b in ag){if(ag.hasOwnPropert
{aq=ag[b];if(aq&&aq.$isFunction&&!aq.$isClass&&aq!==Ext.emptyFn&&aq!==Ext.identityFn){v[b]=r=aq;r.$owner=v;r.$name=b}v[b]=aq}}}if(at.inheritableStatics){v.addInheritableS
at.inheritableStatics}if(j.onClassExtended){a.onExtended(j.onClassExtended,a);delete j.onClassExtended}if(at.platformConfig&&J){J.call(Ext.Class,v,at);delete at.platformCo[illegible]nfig&&P)
{P.call(Ext.Class,v,at);delete at.cachedConfig}if(at.deprecated&&C){C.call(Ext.ClassManager,al,v,at)}s.onBeforeCreated(v,s.data,s);for(d=0,h=an&&an.length;d<h;++d){v.mixin.apply(v,an[d])}for(d=0,h=c.length;d<h;d++){k=c[d];y.setAlias?
y.setAlias(v,k):y.addAlias(v,k)}if(at.singleton){a=new v()}if(!(p instanceof Array)){p=[p]}m=y.getName(a);for(d=0,q=p.length;d<q;d++){g=p[d];y.classes[g]=a;if(H){y.addAlternate(v,g)}else{if(m){o[g]=m;p=ah[m]||(ah[m]=
[]);p.push(g)}}}for(d=0,h=l.length;d<h;d+=2){ao=l[d];if(!ao){ao=e}ao[l[d+1]]=a}y.classes[al]=a;if(!H){if(m&&m!==al){o[al]=m;p=ah[m]||(ah[m]=[]);p.push(al)}}delete j.alternateClassName;if(s.onCreated){s.onCreated.call(a,a)}if(al)
{y.triggerCreated(al)}return a};L.derive=z}(Ext.cmd={}));var Ext=Ext||{};(function(){var C=this,x=Object.prototype,B=x.toString,p=["valueOf","toLocaleString","toString","constructor"],s=function(){},y=function(){},v=function(a){return
a},q=function(){var a=q.caller.caller;return a.$owner.prototype[a.$name].apply(this,arguments)},D=Ext.manifest||{},u,A=/\[object\s*(?:Array|Arguments|\w*Collection|\w*List|HTML\s+document\.all\s+class)\]/,z=/^\\?\/Date\(([-+])?(\d+)(?:
[+-]\d{4})?\)\\?\/$/;Ext.global=C;Ext.now=Date.now||(Date.now=function(){return +new Date()});Ext.ticks=(C.performance&&C.performance.now)?function(){return
performance.now()}:Ext.now;Ext._startTime=Ext.ticks();s.$nullFn=v.$nullFn=s.$emptyFn=v.$identityFn=y.$nullFn=true;y.$privacy="framework";s.$noClearOnDestroy=v.$noClearOnDestroy=true;y.$noClearOnDestroy=true;Ext.suspendLayouts=Ext.resumeLayou
ts=s;for(u in {toString:1}){p=null}Ext.enumerables=p;Ext.apply=function(c,d,a){if(a){Ext.apply(c,a)}if(c&&d&&typeof d==="object"){var b,e,g;for(b in d){c[b]=d[b]}if(p){for(e=p.length;e--;){g=p[e];if(d.hasOwnProperty(g)){c[g]=d[g]}}}}return
c};function r(b,e,a){var d,c;for(d in a){if(a.hasOwnProperty(d)){c=a[d];if(typeof c==="function"){c.$name=d;c.$owner=e;c.$previous=b.hasOwnProperty(d)?b[d]:q}b[d]=c}}}Ext.buildSettings=Ext.apply({baseCSSPrefix:"x-"},Ext.buildSettings||
{});Ext.apply(Ext,{idSeed:0,idPrefix:"ext-
",isSecure:/^https/i.test(window.location.protocol),enableGarbageCollector:false,enableListenerCollection:true,name:Ext.sandboxName||"Ext",privateFn:y,emptyFn:s,identityFn:v,frameStartTime:Ext.now(),manifest:D,enableAria:true,startsWithHashR
e:/^#/,validIdRe:/^[a-z_][a-z0-9\-_]*$/i,BLANK_IMAGE_URL:"data:image/gif;base64,R0lGODlhAQABAID/AMDAwAAAACH5BAEAAAAALAAAAAABAAEAAAICRAEAOw==",makeIdSelector:function(a){return"#"+a},id:function(b,c){if(b&&b.id){return b.id}var a=
(c||Ext.idPrefix)+(++Ext.idSeed);if(b){b.id=a}return a},returnId:function(a){return a.getId()},returnTrue:function(){return true},emptyString:new String(),baseCSSPrefix:Ext.buildSettings.baseCSSPrefix,$eventNameMap:
{},$vendorEventRe:/^(DOMMouse|Moz.+|MS.+|webkit.+)/,canonicalEventName:function(a){return Ext.$eventNameMap[a]||(Ext.$eventNameMap[a]=(Ext.$vendorEventRe.test(a)?a:a.toLowerCase()))},applyIf:function(b,c){var a;if(b){for(a in c)
{if(b[a]===undefined){b[a]=c[a]}}}return b},destroy:function(){var a=arguments.length,b,c;for(b=0;b<a;b++){c=arguments[b];if(c){if(Ext.isArray(c)){this.destroy.apply(this,c)}else{if(Ext.isFunction(c.destroy)){c.destroy()}}}}return
null},destroyMembers:function(c){for(var a,d,b=1,e=arguments,g=e.length;b<g;b++){a=c[d=e[b]];if(a!=null){c[d]=Ext.destroy(a)}}},override:function(b,a){if(b.$isClass){b.override(a)}else{if(typeof b==="function")
{Ext.apply(b.prototype,a)}else{var d=b.self,c;if(d&&d.$isClass){c=a.privates;if(c){a=Ext.apply({},a);delete a.privates;r(b,d,c)}r(b,d,a)}else{Ext.apply(b,a)}}}return b},valueFrom:function(a,c,b){return Ext.isEmpty(a,b)?
c:a},isEmpty:function(b,a){return(b==null)||(!a?b==="":false)||(Ext.isArray(b)&&b.length===0)},isArray:("isArray" in Array)?Array.isArray:function(a){return B.call(a)==="[object Array]"},isDate:function(a){return B.call(a)==="[object
Date]"},isMSDate:function(a){if(!Ext.isString(a)){return false}return z.test(a)},isObject:(B.call(null)==="[object Object]")?function(a){return a!==null&&a!==undefined&&B.call(a)==="[object Object]"&&a.ownerDocument===undefined}:function(a)
{return B.call(a)==="[object Object]"},isSimpleObject:function(a){return a instanceof Object&&a.constructor===Object},isPrimitive:function(a){var b=typeof a;return b==="string"||b==="number"||b==="boolean"},isFunction:(typeof
document!=="undefined"&&typeof document.getElementsByTagName("body")==="function")?function(a){return !!a&&B.call(a)==="[object Function]"}:function(a){return !!a&&typeof a==="function"},isNumber:function(a){return typeof
a==="number"&&isFinite(a)},isNumeric:function(a){return !isNaN(parseFloat(a))&&isFinite(a)},isString:function(a){return typeof a==="string"},isBoolean:function(a){return typeof a==="boolean"},isElement:function(a){return a?
a.nodeType===1:false},isTextNode:function(a){return a?a.nodeName==="#text":false},isDefined:function(a){return typeof a!=="undefined"},isIterable:function(a){if(!a||typeof a.length!=="number"||typeof a==="string"||Ext.isFunction(a)){return
false}if(!a.propertyIsEnumerable){return !!a.item}if(a.hasOwnProperty("length")&&!a.propertyIsEnumerable("length")){return true}return A.test(B.call(a))},isDebugEnabled:s,clone:function(a,c){if(a===null||a===undefined){return
a}if(c!==false&&a.nodeType&&a.cloneNode){return a.cloneNode(true)}var b=B.call(a),d,g,h,j,e;if(b==="[object Date]"){return new Date(a.getTime())}if(b==="[object Array]"){d=a.length;j=[];while(d--){j[d]=Ext.clone(a[d],c)}}else{if(b==="[object
```

Evidence notes:
Image shows hardcoded credentials no longer present.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Medium (5.5) = (Likelihood (6 + 5) /2 = Medium (5.5) + Impact (6 + 5) /2 = Medium (5.5)) /2

Reference(s):

https://www.ftc.gov/tips-advice/business-center/guidance/protecting-personal-information-guide-business
http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf

CVSS:

(AV:N/AC:L/Au:S/C:C/I:P/A:N)

[Back to Top]

[THIS FINDING HAS BEEN REMEDIATED]

Finding(s)

3. JavaScript Library Contains Known Vulnerabilities | Medium (5.25)

Description:

JavaScript libraries are files that contain pre-written JavaScript code which allow for easier development of applications. One, or more, of the libraries identified in the application(s) are known to contain vulnerable code. This can include code inject, information disclosure, and other known security flaws.

Impact:

By using these JavaScript libraries, vulnerabilities are being introduced to the application. This can allow for unauthenticated access to the application and the data it handles.

Test(s) Conducted:

By walking through the application we are able to identify the JavaScript libraries that are being used, including the version. The version is checked against known vulnerabilities.

Finding Comments:

RedTeam identified older versions of jQuery and Sencha Ext JS with known vulnerabilities being utilized by the application.

JQuery 1.9.0 is vulnerable to several cross site scripting vulnerabilities. For more on those, see:
https://www.cvedetails.com/cve/CVE-2016-7103/
https://nvd.nist.gov/vuln/detail/CVE-2015-9251

Sencha Ext JS 4 to 6 before 6.6.0 is vulnerable to XSS attacks, even when passed HTML-escaped data. See:
https://nvd.nist.gov/vuln/detail/CVE-2018-8046

Update: 7/31/2020
RedTeam noted during retesting that the JavaScript library JQuery had been updated and that Sencha Ext had not.

Recommendations:

The JavaScript libraries should be upgraded to the latest vendor recommended version. This update will depend on the libraries that are being used.

Affected System(s):

JQuery:
acesectest.adeptia.com/jquery/jquery.min.js

Sencha references in:
acesectest.adeptia.com/app.js

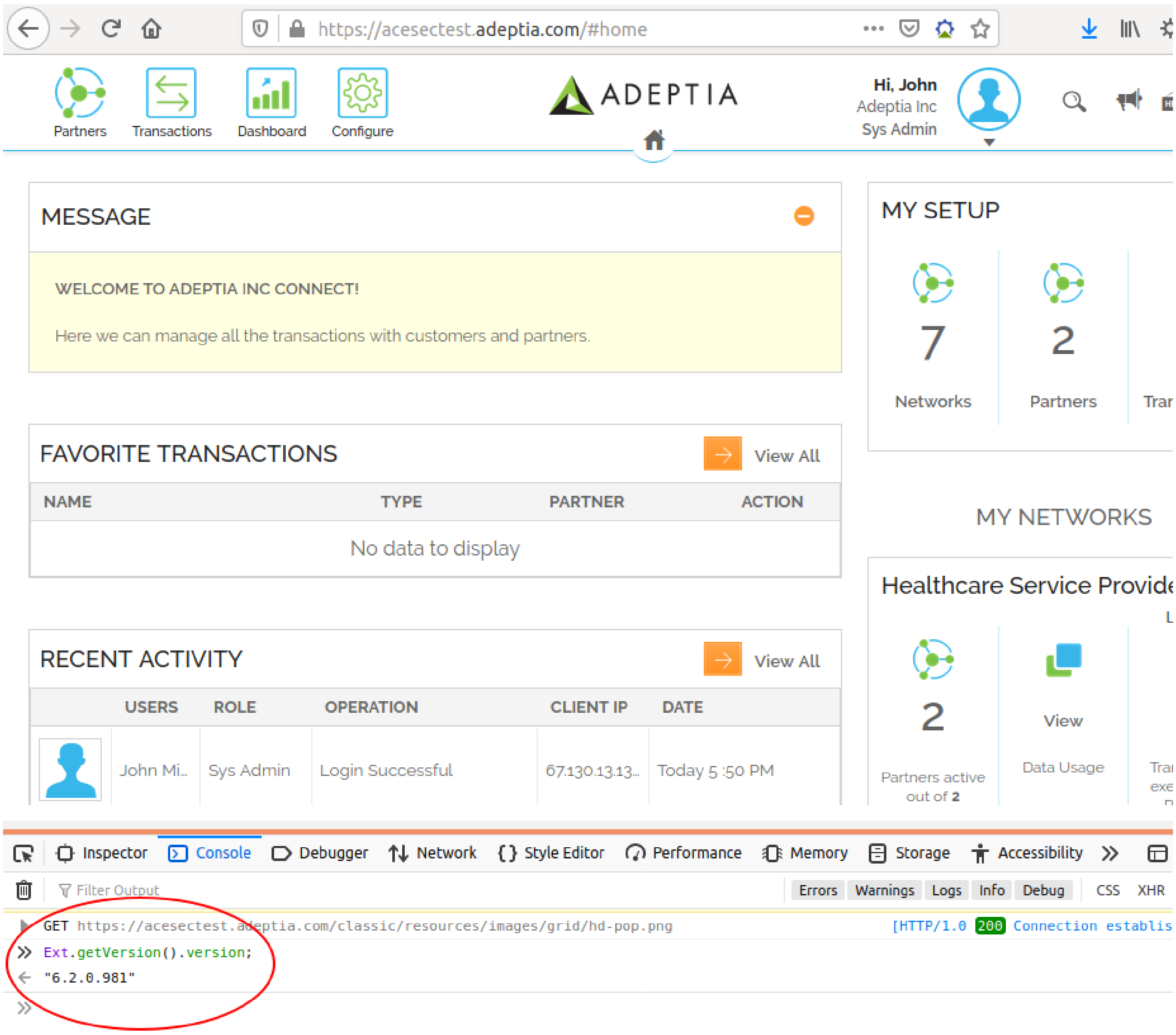
Instance(s):

2

Status:

Not Remediated

Evidence:



MESSAGE

WELCOME TO ADEPTIA INC CONNECT!

Here we can manage all the transactions with customers and partners.

FAVORITE TRANSACTIONS View All

NAME	TYPE	PARTNER	ACTION
No data to display			

RECENT ACTIVITY View All

	USERS	ROLE	OPERATION	CLIENT IP	DATE
	John Mi...	Sys Admin	Login Successful	67.130.13.13...	Today 5 :50 PM

MY SETUP

7 Networks

2 Partners

MY NETWORKS

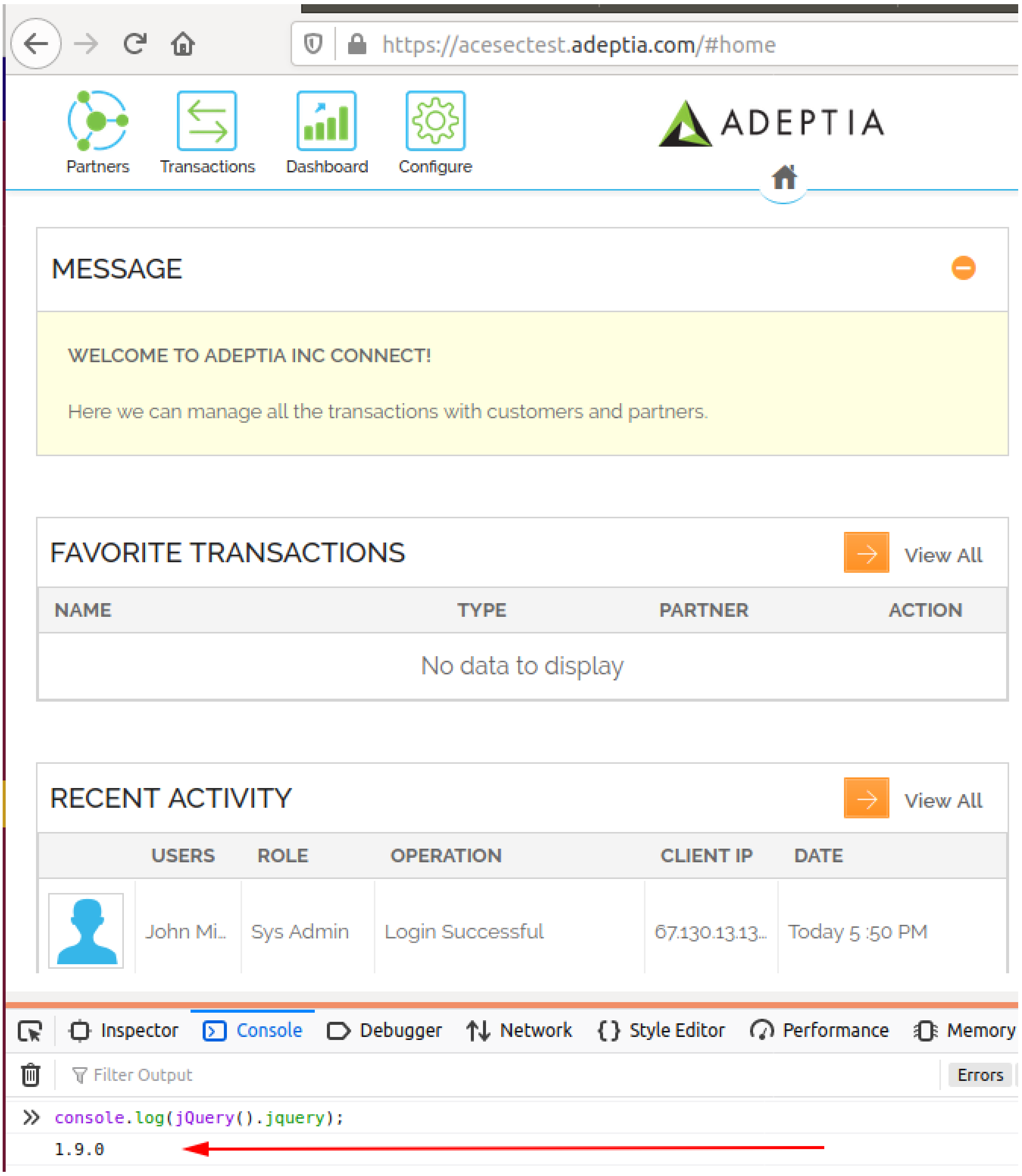
Healthcare Service Provide

2 Partners active out of **2**

View Data Usage

```
GET https://acesectest.adeptia.com/classic/resources/images/grid/hd-pop.png    [HTTP/1.0 200 Connection establis
>> Ext.getVersion().version;
<- "6.2.0.981"
>>
```

Evidence notes:
The image shows Ext JS version.


https://acesectest.adeptia.com/#home

Partners Transactions Dashboard Configure

ADEPTIA

MESSAGE

WELCOME TO ADEPTIA INC CONNECT!

Here we can manage all the transactions with customers and partners.

FAVORITE TRANSACTIONS View All

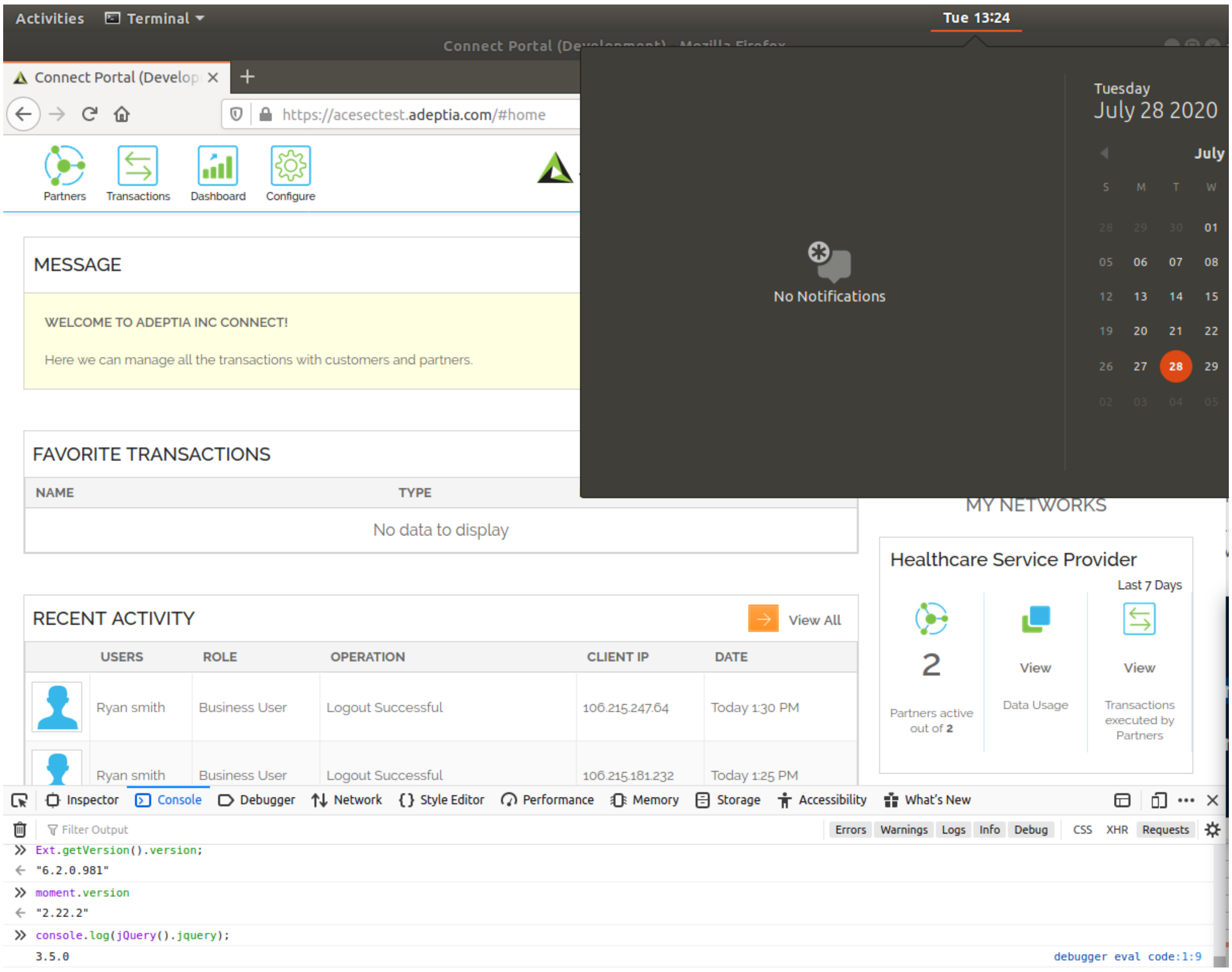
NAME	TYPE	PARTNER	ACTION
No data to display			

RECENT ACTIVITY View All

	USERS	ROLE	OPERATION	CLIENT IP	DATE
	John Mi...	Sys Admin	Login Successful	67.130.13.13...	Today 5 :50 PM

```
>> console.log(jQuery().jquery);
   1.9.0
```

Evidence notes:
Image shows version of JQuery in use.



MESSAGE

WELCOME TO ADEPTIA INC CONNECT!

Here we can manage all the transactions with customers and partners.

FAVORITE TRANSACTIONS

NAME	TYPE
No data to display	

MY NETWORKS

Healthcare Service Provider

Last 7 Days

2 Partners active out of **2** — View Data Usage — View Transactions executed by Partners

RECENT ACTIVITY — View All

USERS	ROLE	OPERATION	CLIENT IP	DATE
Ryan smith	Business User	Logout Successful	106.215.247.64	Today 1:30 PM
Ryan smith	Business User	Logout Successful	106.215.181.232	Today 1:25 PM

```
>> Ext.getVersion().version;
<- "6.2.0.981"
>> moment.version
<- "2.22.2"
>> console.log(jQuery().jquery);
   3.5.0                                                    debugger eval code:1:9
```

Evidence notes:
Image shows the JavaScript libraries in use.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood***(*Threat Agents + Vulnerability Factors) /2 +* ***Impact***(*Technical Impact + Business Impact) /2 =* ***Risk Rating***(*Likelihood + Impact) /2*

Medium (5.25) = (Likelihood (5 + 5) /2 = Medium (5) + Impact (6 + 5) /2 = Medium (5.5)) /2

Reference(s):

https://owasp.org/www-project-top-ten/
https://www.acunetix.com/vulnerabilities/web/vulnerable-javascript-library

CVSS:

(AV:N/AC:M/Au:N/C:P/I:P/A:P)

[Back to Top]

Finding(s)

4. Diffie-Hellman Modulus <= 1024bits (Logjam) | Medium (3.5)

[THIS FINDING HAS BEEN REMEDIATED]

Description:

The remote system(s) allows secure socket layer (SSL) or transport layer security (TLS) connections with one or more Diffie-Hellman (DH) modules utilizing a key length of 1024 bits or less. Diffie-Hellman is an algorithm used to establish a shared secret between two parties that are creating a secure connection. The algorithm is flawed in the sense that, depending on the bit length, an individual may be able to crack the shared secret and decrypt communication.

Impact:

Through cryptanalysis, a third party may be able to find the shared secret in a short amount of time depending on the bit length of the DH mode. It is estimated that a single user can compute a 512-bit prime, an academic team can compute a 768-bit prime, and a nation-state can compute a 1024-bit prime; thus using that prime to decrypt data and potentially view sensitive information.

Test(s) Conducted:

Several connections are made to the affected system, identifying the supported ciphers. This information shows that Diffie-Hellman ciphers, which are equal to or less than 1024 bits in length, are being utilized.

Finding Comments:

During testing, RedTeam identified hosts that are capable of negotiating encrypted communications using Diffie-Hellman ciphers with key lengths less than or equal to 1024 bits. Due to mathematical biases in the Diffie-Hellman algorithm, these shorter key lengths may be able to be discovered by an attacker. If discovered, the attacker can utilize these keys to decrypt intercepted network traffic.

Update: 7/31/2020
RedTeam noted during retesting that the application no longer accepted ciphers vulnerable to LOGJAM.

Recommendations:

Diffie-Hellman ciphers should be reconfigured to use a key length of 2048 or higher.

Affected System(s):

acesectest.adeptia.com

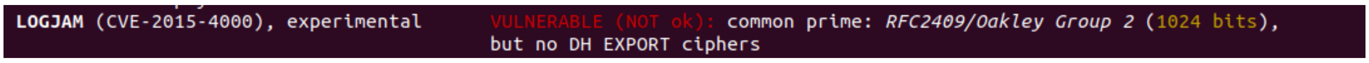
Instance(s):

1

Status:

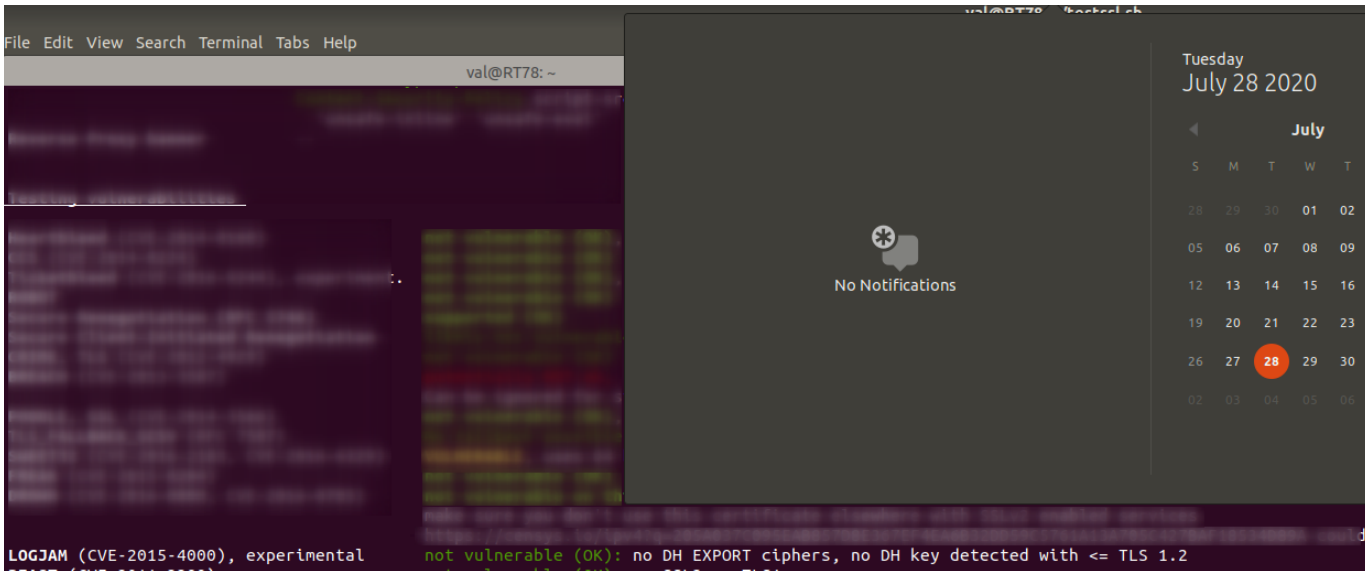
Remediated

Evidence:



```
LOGJAM (CVE-2015-4000), experimental        VULNERABLE (NOT ok): common prime: RFC2409/Oakley Group 2 (1024 bits),
                                            but no DH EXPORT ciphers
```

Evidence notes:
The image shows testssl results with LOGJAM vulnerability.



[redacted]

```
LOGJAM (CVE-2015-4000), experimental    not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
```

Evidence notes:
Image shows the application does not accept ciphers vulnerable to LOGJAM.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Medium (3.5) = (Likelihood (3 + 4) /2 = Medium (3.5) + Impact (4 + 3) /2 = Medium (3.5)) /2

Reference(s):

https://weakdh.org/
https://access.redhat.com/articles/1456263

CVSS:

(AV:N/AC:M/Au:N/C:P/I:P/A:N)

[Back to Top]

[THIS FINDING HAS BEEN REMEDIATED]

Finding(s)

5. Authentication Bypass | Medium (3.5)

[THIS FINDING HAS BEEN REMEDIATED]

Description:

Authentication is used with a wide variety of services and applications in order to prevent unauthorized access to sensitive information and functionality. In this case the implemented authentication scheme was able to be bypassed, and unauthorized access was gained to sensitive information and/or functionality.

Impact:

An attacker that is able to bypass authentication may be able to view or modify application or service functionality, or access information that is meant for a restricted group of users, without possessing valid credentials.

Test(s) Conducted:

Authentication bypass methods are dependent on the authentication scheme and implementation. Common methods include navigating directly to protected functionality in a web browser or using SQL injection to bypass username or password verification.

Finding Comments:

While testing the application, RedTeam tampered with the SAML assertion in various ways. The application caught all of the tampered attempts and returned an error for them, however, in some cases, the application would continue to retry the authentication. A tampered request would, under specific circumstances, lead to an active session.

This was evident after receiving the error, RedTeam closed the window and opened a new one to acesectest.adeptia.com. A session would be automatically opened without the user needing to send in credentials a second time. This is of particular concern with a shared computer or one that someone might be able to obtain access to. In a browser with settings that reopen all earlier tabs, or where the application was the home page, this would automatically open a page to an open session. RedTeam would like to note that there is a limited window of time in which this will be successful, and the risk rating for this finding has been reduced accordingly. When an application has detected a tampered request in the process of authentication, it should halt it entirely and start the process over.

During testing, these calls were found to be of particular interest:
acesectest.adeptia.com/saml/SSO
dev-918046.okta.com/app/adeptiadev918046_samlsecurity2_1/exk4kbyzs6WXkSBcm357/sso/saml
acesectest.adeptia.com/SAMLErrorHandler?mode=1234
as well as others that sent SAML data in the body of a POST request as a redirect parameter. There is one instance of this finding, these URLs are given in the case that they might be useful in observing and debugging the process.

Update: 7/31/2020
RedTeam noted during retesting that the actions of the authentication process were changed. The application did not try to reattempt login if the SAML data had been tampered with and this finding can not be reproduced.

Recommendations:

In vendor products, make sure that the most recent security patches are applied up to the current date and follow the documentation to secure the product deployment. If the service or application is custom developed, modify the codebase to prevent the specific technique that was deployed to bypass authentication. The use of an authentication middleware is recommended as a defense-in-depth measure.

Affected System(s):

acesectest.adeptia.com

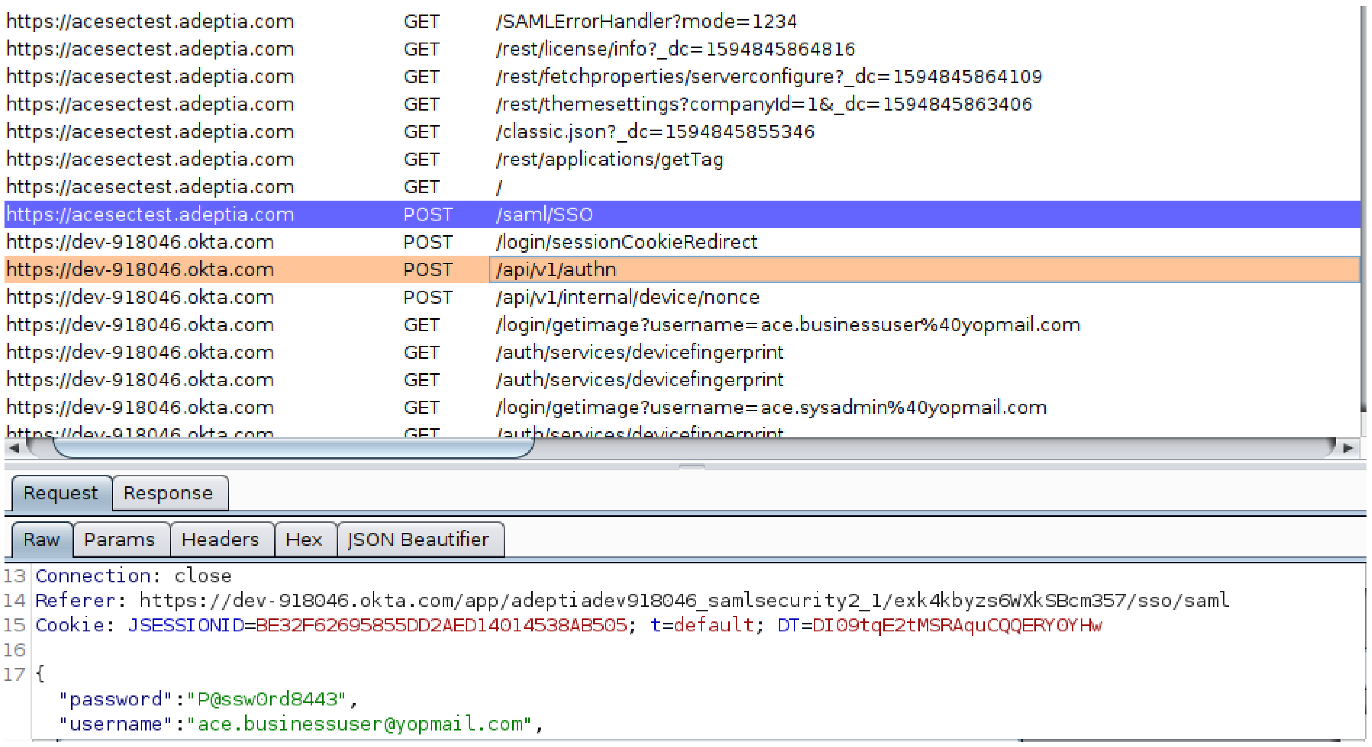
Instance(s):

1

Status:

Remediated

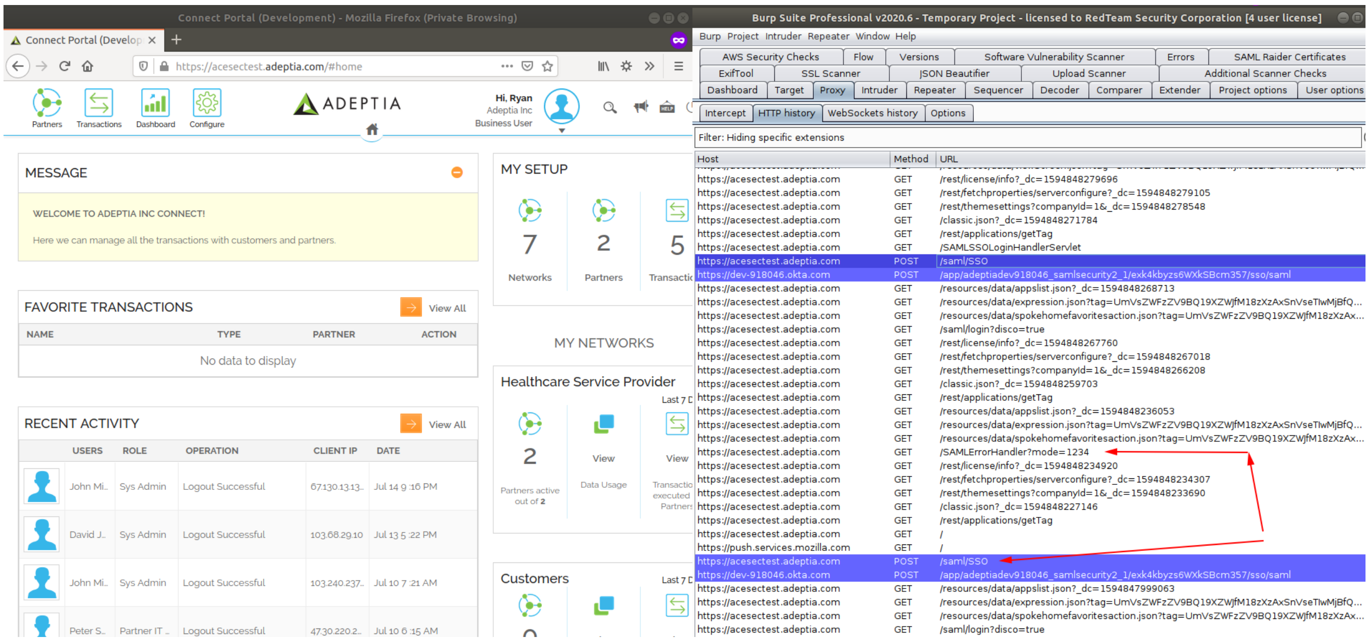
Evidence:


```
https://acesectest.adeptia.com    GET    /SAMLErrorHandler?mode=1234
https://acesectest.adeptia.com    GET    /rest/license/info?_dc=1594845864816
https://acesectest.adeptia.com    GET    /rest/fetchproperties/serverconfigure?_dc=1594845864109
https://acesectest.adeptia.com    GET    /rest/themesettings?companyId=1&_dc=1594845863406
https://acesectest.adeptia.com    GET    /classic.json?_dc=1594845855346
https://acesectest.adeptia.com    GET    /rest/applications/getTag
https://acesectest.adeptia.com    GET    /
https://acesectest.adeptia.com    POST   /saml/SSO
https://dev-918046.okta.com       POST   /login/sessionCookieRedirect
https://dev-918046.okta.com       POST   /api/v1/authn
https://dev-918046.okta.com       POST   /api/v1/internal/device/nonce
https://dev-918046.okta.com       GET    /login/getimage?username=ace.businessuser%40yopmail.com
https://dev-918046.okta.com       GET    /auth/services/devicefingerprint
https://dev-918046.okta.com       GET    /auth/services/devicefingerprint
https://dev-918046.okta.com       GET    /login/getimage?username=ace.sysadmin%40yopmail.com
https://dev-918046.okta.com       GET    /auth/services/devicefingerprint
```

```
Connection: close
Referer: https://dev-918046.okta.com/app/adeptiadev918046_samlsecurity2_1/exk4kbyzs6WXkSBcm357/sso/saml
Cookie: JSESSIONID=BE32F62695855DD2AED14014538AB505; t=default; DT=DI09tqE2tMSRAquCQQERY0YHw

{
  "password":"P@ssw0rd8443",
  "username":"ace.businessuser@yopmail.com",
```

Evidence notes:
The above image shows the proper, initial exchange of the username and password. The orange highlighted line is the POST request with credentials, with the details in the bottom pane. The purple line is a subsequent, tampered SAML exchange.



```
Host                              Method  URL
https://acesectest.adeptia.com    GET     /rest/license/info?_dc=1594848279696
https://acesectest.adeptia.com    GET     /rest/fetchproperties/serverconfigure?_dc=1594848279105
https://acesectest.adeptia.com    GET     /rest/themesettings?companyId=1&_dc=1594848278548
https://acesectest.adeptia.com    GET     /classic.json?_dc=1594848271784
https://acesectest.adeptia.com    GET     /rest/applications/getTag
https://acesectest.adeptia.com    GET     /SAMLSSOLoginHandlerServlet
https://acesectest.adeptia.com    POST    /saml/SSO
https://dev-918046.okta.com       POST    /app/adeptiadev918046_samlsecurity2_1/exk4kbyzs6WXkSBcm357/sso/saml
https://acesectest.adeptia.com    GET     /resources/data/appslist.json?_dc=1594848268713
https://acesectest.adeptia.com    GET     /resources/data/expression.json?tag=UmVsZWFzZV9BQ19XZWJfM18zXzAxSnVseTIwMjBfQ...
https://acesectest.adeptia.com    GET     /resources/data/spokehomefavoritesaction.json?tag=UmVsZWFzZV9BQ19XZWJfM18zXzAx...
https://acesectest.adeptia.com    GET     /saml/login?disco=true
https://acesectest.adeptia.com    GET     /rest/license/info?_dc=1594848267760
https://acesectest.adeptia.com    GET     /rest/fetchproperties/serverconfigure?_dc=1594848267018
https://acesectest.adeptia.com    GET     /rest/themesettings?companyId=1&_dc=1594848266208
https://acesectest.adeptia.com    GET     /classic.json?_dc=1594848259703
https://acesectest.adeptia.com    GET     /rest/applications/getTag
https://acesectest.adeptia.com    GET     /resources/data/appslist.json?_dc=1594848236053
https://acesectest.adeptia.com    GET     /resources/data/expression.json?tag=UmVsZWFzZV9BQ19XZWJfM18zXzAxSnVseTIwMjBfQ...
https://acesectest.adeptia.com    GET     /resources/data/spokehomefavoritesaction.json?tag=UmVsZWFzZV9BQ19XZWJfM18zXzAx...
https://acesectest.adeptia.com    GET     /SAMLErrorHandler?mode=1234
https://acesectest.adeptia.com    GET     /rest/license/info?_dc=1594848234920
https://acesectest.adeptia.com    GET     /rest/fetchproperties/serverconfigure?_dc=1594848234307
https://acesectest.adeptia.com    GET     /rest/themesettings?companyId=1&_dc=1594848233690
https://acesectest.adeptia.com    GET     /classic.json?_dc=1594848227146
https://acesectest.adeptia.com    GET     /rest/applications/getTag
https://acesectest.adeptia.com    GET     /
https://push.services.mozilla.com GET     /
https://acesectest.adeptia.com    POST    /saml/SSO
https://dev-918046.okta.com       POST    /app/adeptiadev918046_samlsecurity2_1/exk4kbyzs6WXkSBcm357/sso/saml
https://acesectest.adeptia.com    GET     /resources/data/appslist.json?_dc=1594847999063
https://acesectest.adeptia.com    GET     /resources/data/expression.json?tag=UmVsZWFzZV9BQ19XZWJfM18zXzAxSnVseTIwMjBfQ...
https://acesectest.adeptia.com    GET     /resources/data/spokehomefavoritesaction.json?tag=UmVsZWFzZV9BQ19XZWJfM18zXzAx...
https://acesectest.adeptia.com    GET     /saml/login?disco=true
```

Connect Portal (Development) - Mozilla Firefox (Private Browsing)
Burp Suite Professional v2020.6 - Temporary Project - licensed to RedTeam Security Corporation [4 user license]
https://acesectest.adeptia.com/#home
Partners
Transactions
Dashboard
Configure
ADEPTIA
Hi, Ryan
Adeptia Inc
Business User
MESSAGE
WELCOME TO ADEPTIA INC CONNECT!
Here we can manage all the transactions with customers and partners.
FAVORITE TRANSACTIONS
View All
No data to display
RECENT ACTIVITY
MY SETUP
7 Networks
2 Partners
5 Transactions
MY NETWORKS
Healthcare Service Provider
Customers
Burp Project Intruder Repeater Window Help
AWS Security Checks
Flow
Versions
Software Vulnerability Scanner
Errors
SAML Raider Certificates
ExifTool
SSL Scanner
JSON Beautifier
Upload Scanner
Additional Scanner Checks
Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Extender
Project options
User options
Intercept
HTTP history
WebSockets history
Options
Filter: Hiding specific extensions

Evidence notes:
Subsequently:
The bottom purple bands indicate tampered SAML traffic. Following above to the red arrows is the call to acesectest.adeptia.com/SAMLErrorHandler?mode=1234, where the system appropriately caught the tampering and responded with an error. Note in the section between the two bands of purple, a new request with credentials (dev-918046.okta.com/api/v1/authn) is not sent. (This can be seen in the other image, in the orange band of traffic.) At the higher purple-banded traffic, the system retried the SAML exchange, and the traffic above it is in session.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Medium (3.5) = (Likelihood (4 + 3) /2 = Medium (3.5) + Impact (4 + 3) /2 = Medium (3.5)) /2

Reference(s):

https://www.owasp.org/index.php/Testing_for_Bypassing_Authentication_Schema_(OTG-AUTHN-004)
https://cyware.com/news/authentication-bypass-vulnerability-what-is-it-and-how-to-stay-protected-ccc2ea38

CVSS:

7.5 (AV:N/AC:L/Au:N/C:P/I:P/A:P)

[Back to Top]

[THIS FINDING HAS BEEN REMEDIATED]

Finding(s)

6. Cookie Missing HTTPOnly Flag | Medium (3.5)

Description:

An instance was discovered where cookies were not secured by using the HTTPOnly attribute. The HTTPOnly attribute ensures that the cookies value cannot be read by client-side JavaScript and can prevent attacks like cross-site scripting.

Impact:

An attacker could leverage this weakness and potentially steal sensitive information, including session information, and possibly enable unauthorized access to the application and it’s data.

Test(s) Conducted:

We examined cookies looking for the HTTPOnly attribute setting.

Finding Comments:

While analyzing the cookies presented by the application, several were discovered which do not contain an HttpOnly flag. The HttpOnly flag enables functionality in the browser which prevents the disclosure of cookie values through attacker-controlled methods, such as Cross-Site Scripting. The cookies in question were ACCESS_TOKEN and saml-user-attributes.

Recommendations:

We recommend that the HTTPOnly attribute be set on all cookies, especially those that transmit sensitive information. If cookies are used to transmit session tokens, application functionality that is accessed over HTTPS should use its own session handling mechanism, separate from all HTTP communications, and the session tokens used should never be transmitted over an unencrypted channel.

Affected System(s):

acesectest.adeptia.com/SAMLSSOLoginHandlerServlet ACCESS_TOKEN
acesectest.adeptia.com/SAMLSSOLoginHandlerServlet saml-user-attributes

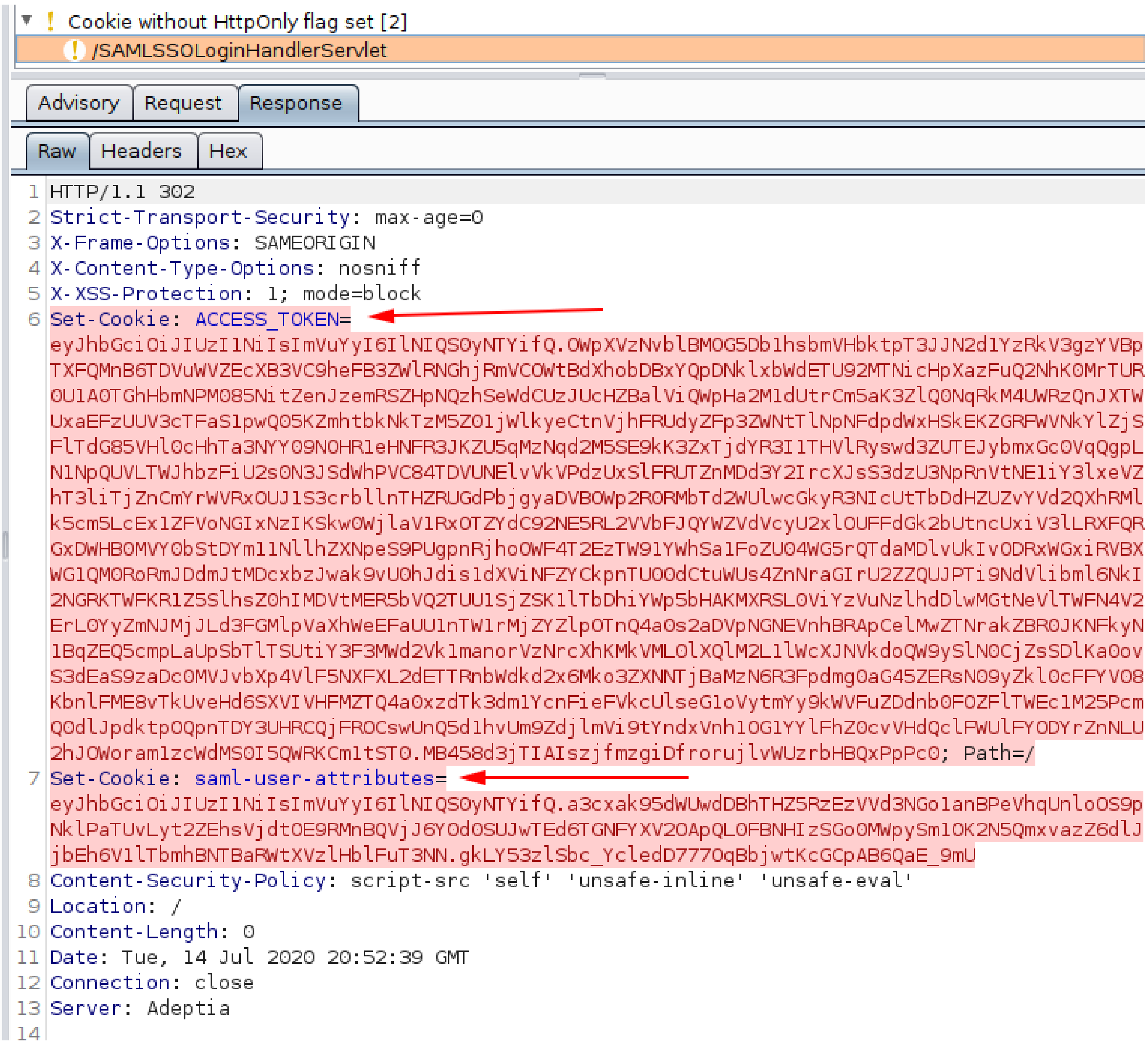
Instance(s):

2

Status:

Not Remediated

Evidence:


Cookie without HttpOnly flag set [2]
/SAMLSSOLoginHandlerServlet
Advisory
Request
Response
Raw
Headers
Hex

```
HTTP/1.1 302
Strict-Transport-Security: max-age=0
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Set-Cookie: ACCESS_TOKEN=
eyJhbGciOiJIUzI1NiIsImVuYyI6IlNIQS0yNTYifQ.OWpXVzNvblBMOG5Db1hsbmVHbktpT3JJN2d1YzRkV3gzYVBp
TXFQMnB6TDVuWVZEcXB3VC9heFB3ZWlRNGhjRmVCOWtBdXhobDBxYQpDNklxbWdETU92MTNicHpXazFuQ2NhK0MrTUR
OU1AOTGhHbmNPM085NitZenJzemRSZHpNQzhSeWdCUzJUcHZBalViQWpHa2M1dUtrCm5aK3ZlQ0NqRkM4UWRzQnJXTW
UxaEFzUUV3cTFaS1pwQ05KZmhtbkNkTzM5Z01jWlkyeCtnVjhFRUdyZFp3ZWNtTlNpNFdpdWxHSkEKZGRFWVNkYlZjS
FlTdG85VHl0cHhTa3NYY09NOHR1eHNFR3JKZU5qMzNqd2M5SE9kK3ZxTjdYR3I1THVlRyswd3ZUTEJybmxGc0VqQgpL
N1NpQUVLTWJhbzFiU2s0N3JSdWhPVC84TDVUNElvVkVPdzUxSlFRUTZnMDd3Y2IrcXJsS3dzU3NpRnVtNE1iY3lxeVZ
hT3liTjZnCmYrWVRxOUJ1S3crbllnTHZRUGdPbjgyaDVBOWp2R0RMbTd2WUlwcGkyR3NIcUtTbDdHZUZvYVd2QXhRMl
k5cm5LcEx1ZFVoNGIxNzIKSkwOWjlaV1RxOTZYdC92NE5RL2VVbFJQYWZVdVcyU2xlOUFFdGk2bUtncUxiV3lLRXFQR
GxDWHBOMVY0bStDYm11NllhZXNpeS9PUgpnRjhoOWF4T2EzTW91YWhSa1FoZU04WG5rQTdaMDlvUkIvODRxWGxiRVBX
WG1QM0RoRmJDdmJtMDcxbzJwak9vU0hJdis1dXViNFZYCkpnTU00dCtuWUs4ZnNraGIrU2ZZQUJPTi9NdVlibml6NkI
2NGRKTWFKR1Z5SlhsZ0hIMDVtMER5bVQ2TUU1SjZSK1lTbDhiYWp5bHAKMXRSL0ViYzVuNzlhdDlwMGtNeVlTWFN4V2
ErL0YyZmNJMjJLd3FGMlpVaXhWeEFaUU1nTW1rMjZYZlpOTnQ4a0s2aDVpNGNEVnhBRApCelMwZTNrakZBR0JKNFkyN
1BqZEQ5cmpLaUpSbTlTSUtiY3F3Mwd2Vk1manorVzNrcXhKMkVML0lXQlM2L1lWcXJNVkdoQW9ySlN0CjZsSDlKa0ov
S3dEaS9zaDc0MVJvbXp4VlF5NXFXL2dETTRnbWdkd2x6Mko3ZXNNTjBaMzN6R3FpdmgwaG45ZERsN09yZkl0cFFYV08
KbnlFME8vTkUveHd6SXVIVHFMZTQ4a0xzdTk3dm1YcnFieFVkcUlseG1oVytmYy9kWVFuZDdnb0F0ZFlTWEc1M25Pcm
Q0dlJpdktpOQpnTDY3UHRCQjFROCswUnQ5d1hvUm9ZdjlmVi9tYndxVnh1OG1YYlFhZ0cvVHdQclFWUlFYODYrZnNLU
2hJOWoram1zcWdMSOI5QWRKCm1tST0.MB458d3jTIAIszjfmzgiDfrorujlvWUzrbHBQxPpPc0; Path=/
Set-Cookie: saml-user-attributes=
eyJhbGciOiJIUzI1NiIsImVuYyI6IlNIQS0yNTYifQ.a3cxak95dWUwdDBhTHZ5RzEzVVd3NGo1anBPeVhqUnloOS9p
NklPaTUvLyt2ZEhsVjdtOE9RMnBQVjJ6Y0d0SUJwTEd6TGNFYXV2OApQL0FBNHIzSGoOMWpySm1OK2N5QmxvazZ6dlJ
jbEh6V1lTbmhBNTBaRWtXVzlHblFuT3NN.gkLY53zlSbc_YcledD777OqBbjwtKcGCpAB6QaE_9mU
Content-Security-Policy: script-src 'self' 'unsafe-inline' 'unsafe-eval'
Location: /
Content-Length: 0
Date: Tue, 14 Jul 2020 20:52:39 GMT
Connection: close
Server: Adeptia
```

Evidence notes:
The image shows cookies without the HTTPOnly flag.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood***(*Threat Agents + Vulnerability Factors) /2* + ***Impact***(*Technical Impact + Business Impact) /2* = ***Risk Rating***(*Likelihood + Impact) /2*

Medium (3.5) = (Likelihood (3 + 4) /2 = Medium (3.5) + Impact (3 + 4) /2 = Medium (3.5)) /2

Reference(s):

https://www.owasp.org/index.php/HttpOnly

CVSS:

(AV:N/AC:M/Au:N/C:P/I:N/A:N)

[Back to Top]

Finding(s)

7. Verbose Error Message | Low (3.25)

[THIS FINDING HAS BEEN REMEDIATED]

Description:

The software generates an error message that includes sensitive information about its environment, users, or associated data therefore producing an information leak. Revealing system data or debugging information helps an adversary learn about the system and form a plan of attack. An information leak occurs when system data or debugging information leaves the program through an output stream or logging function.

Impact:

Depending upon the system configuration, this information can be dumped to a console, written to a log file, or exposed to a remote user. In some cases the error message tells the attacker precisely what sort of an attack the system will be vulnerable to.

For example, a database error message can reveal that the application is vulnerable to a SQL injection attack. Other error messages can reveal more oblique clues about the system. In some common examples, the search path could imply information about the type of operating system, the applications installed on the system, and the amount of care that the administrators have put into configuring the program.

As a result, all of this information could and would be used by an attacker to launch a more targeted attack.

Test(s) Conducted:

We caused the application to intentionally error out in an effort to examine the content of its error messages. Upon doing so, we took note of the content and examined it for situations where too much information was provided that would aid an attacker with additional information about the system, users, platform, etc.

Finding Comments:

During testing, RedTeam was able to generate errors within the application that presented verbose messages to the user. These error messages can be utilized by attackers to identify the software stack, webserver version, and type and potentially create targeted attacks against the software or server using the information retrieved.

It was possible to generate the first error message with the above GET request only, no prior authentication, the following two calls did require authentication.

Update: 7/31/2020
RedTeam noted during retesting of the application that this finding could no longer be replicated.

Recommendations:

We recommend limiting the content of the error messages to provide on the least amount of information necessary to communicate the nature of the error. We strongly recommend disabling default error message and overly verbose messaging, such as stack traces in order to prevent information leakage.

Affected System(s):

https://acesectest.adeptia.com:8443/adeptia/serviceAction/duplicateValueCheckInDB?entityName=EmployeeBenefitDetails&fieldName=x' OR full_name LIKE '%Bob%&id=010000001218159376425640400011&groupOwner=IndigoGroup:1921680010751481815660483000002&service=DataMapping&operation=create&validationLevel=undefined

https://acesectest.adeptia.com/#transaction/definition/transactionsettings/11594038863749

https://acesectest.adeptia.com/rest/transaction?sequenceId=-1

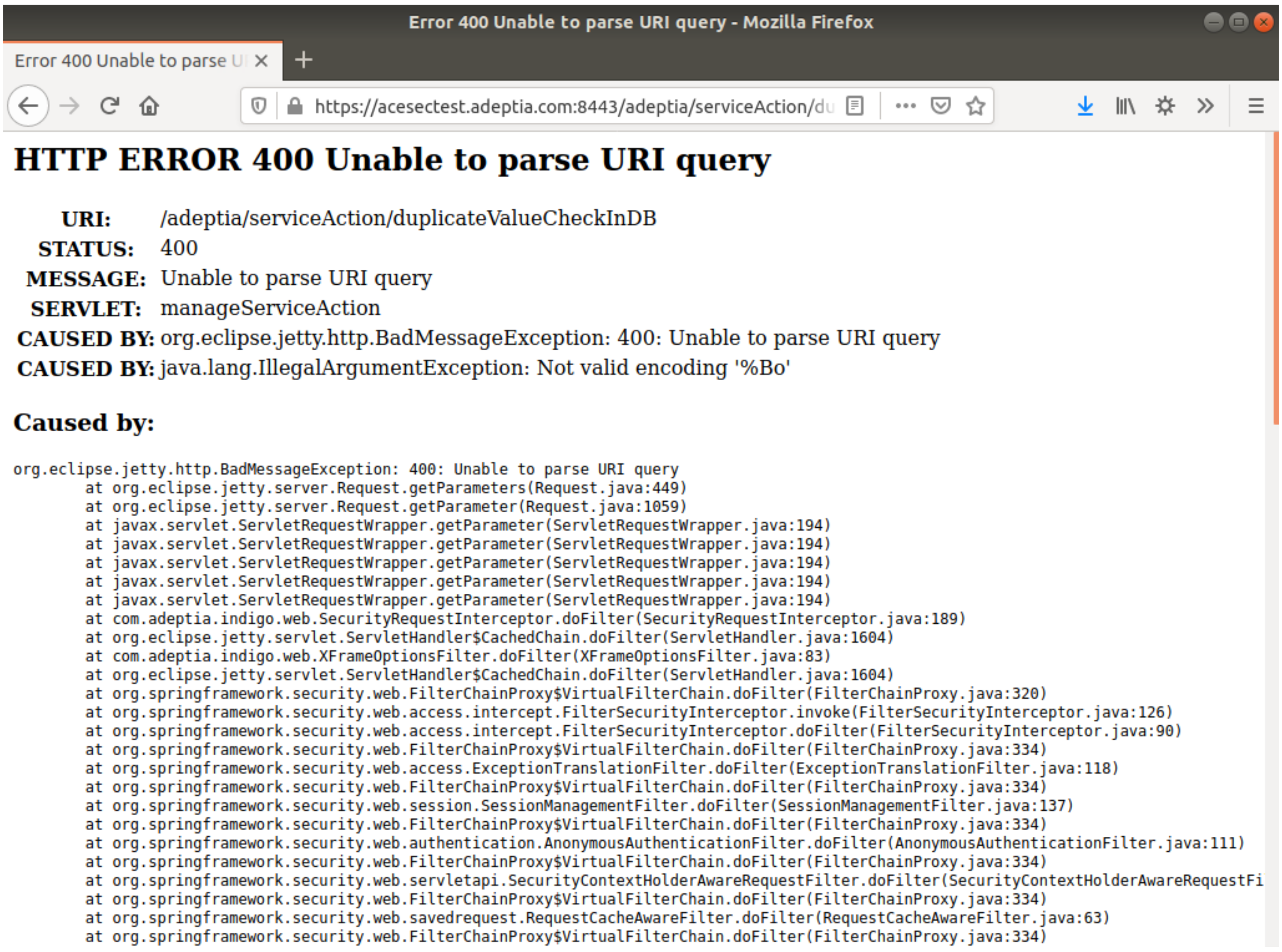
Instance(s):

3

Status:

Remediated

Evidence:


Error 400 Unable to parse URI query - Mozilla Firefox
Error 400 Unable to parse U
https://acesectest.adeptia.com:8443/adeptia/serviceAction/du

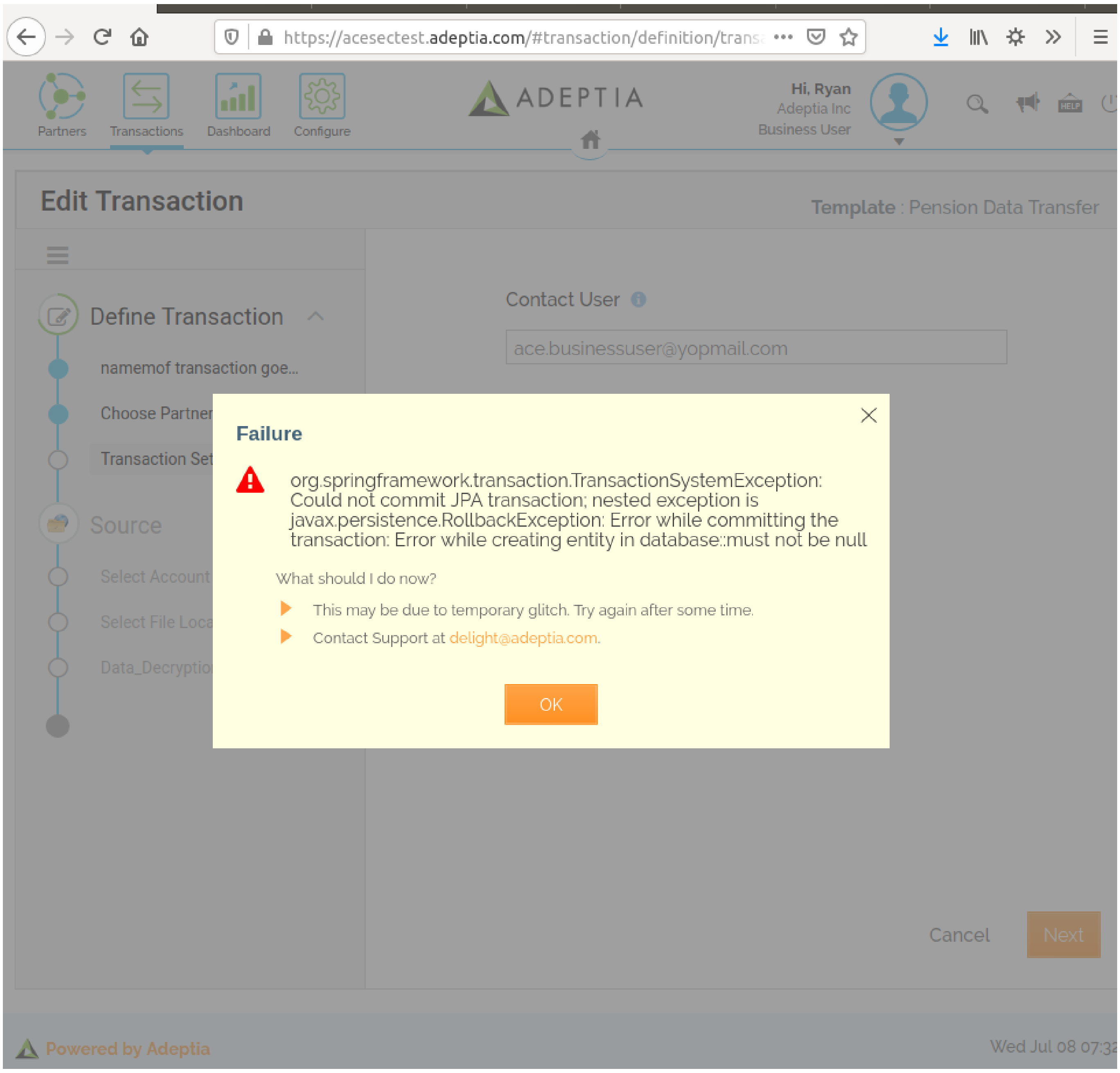
HTTP ERROR 400 Unable to parse URI query

URI: /adeptia/serviceAction/duplicateValueCheckInDB
STATUS: 400
MESSAGE: Unable to parse URI query
SERVLET: manageServiceAction
CAUSED BY: org.eclipse.jetty.http.BadMessageException: 400: Unable to parse URI query
CAUSED BY: java.lang.IllegalArgumentException: Not valid encoding '%Bo'

Caused by:

```
org.eclipse.jetty.http.BadMessageException: 400: Unable to parse URI query
	at org.eclipse.jetty.server.Request.getParameters(Request.java:449)
	at org.eclipse.jetty.server.Request.getParameter(Request.java:1059)
	at javax.servlet.ServletRequestWrapper.getParameter(ServletRequestWrapper.java:194)
	at javax.servlet.ServletRequestWrapper.getParameter(ServletRequestWrapper.java:194)
	at javax.servlet.ServletRequestWrapper.getParameter(ServletRequestWrapper.java:194)
	at javax.servlet.ServletRequestWrapper.getParameter(ServletRequestWrapper.java:194)
	at javax.servlet.ServletRequestWrapper.getParameter(ServletRequestWrapper.java:194)
	at com.adeptia.indigo.web.SecurityRequestInterceptor.doFilter(SecurityRequestInterceptor.java:189)
	at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1604)
	at com.adeptia.indigo.web.XFrameOptionsFilter.doFilter(XFrameOptionsFilter.java:83)
	at org.eclipse.jetty.servlet.ServletHandler$CachedChain.doFilter(ServletHandler.java:1604)
	at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:320)
	at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invoke(FilterSecurityInterceptor.java:126)
	at org.springframework.security.web.access.intercept.FilterSecurityInterceptor.doFilter(FilterSecurityInterceptor.java:90)
	at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:334)
	at org.springframework.security.web.access.ExceptionTranslationFilter.doFilter(ExceptionTranslationFilter.java:118)
	at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:334)
	at org.springframework.security.web.session.SessionManagementFilter.doFilter(SessionManagementFilter.java:137)
	at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:334)
	at org.springframework.security.web.authentication.AnonymousAuthenticationFilter.doFilter(AnonymousAuthenticationFilter.java:111)
	at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:334)
	at org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter.doFilter(SecurityContextHolderAwareRequestFi
	at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:334)
	at org.springframework.security.web.savedrequest.RequestCacheAwareFilter.doFilter(RequestCacheAwareFilter.java:63)
	at org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:334)
```

Evidence notes:
The image shows an error message presented to the user from an unauthenticated GET request.



Failure

org.springframework.transaction.TransactionSystemException: Could not commit JPA transaction; nested exception is javax.persistence.RollbackException: Error while committing the transaction: Error while creating entity in database::must not be null

What should I do now?

- This may be due to temporary glitch. Try again after some time.
- Contact Support at delight@adeptia.com.

OK

Evidence notes:
The image shows a verbose error message.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Low (3.25) = (Likelihood (3 + 3) /2 = Low (3) + Impact (3 + 4) /2 = Medium (3.5)) /2

Reference(s):

https://cwe.mitre.org/data/definitions/209.html

https://www.owasp.org/index.php/Top_10_2007-Information_Leakage_and_Improper_Error_Handling

CVSS:

(AV:N/AC:L/Au:S/C:P/I:N/A:N)

[Back to Top]

[THIS FINDING HAS BEEN REMEDIATED]

Finding(s)

8. Information Disclosure | Low (3)

[THIS FINDING HAS BEEN REMEDIATED]

Description:

During a review of the content served by the application, it was identified the server presents sensitive information. This information can come in a variety of different forms, including: server IP addresses, server names, user information, and many others.

Impact:

This information does not directly make the application or server exploitable, but it does give information to attackers that can allow them to further target systems, services, and users.

Test(s) Conducted:

The server is fingerprinted and crawled to identify any information that can be used to narrow down attack vectors for the system and its users.

Finding Comments:

During testing, it was found that information that had been in the users' session remained on the screen and visible in other tabs, even after the session ended. Attempts to interact with the application or use the back button required login.

Update: 7/31/2020
RedTeam noted during retesting that the application now clears the screen after the session times out.

Recommendations:

Sensitive information should be removed from public access. If the information is needed, it should be presented only after a user has been authenticated.

Affected System(s):

https://acesectest.adeptia.com:8443/adeptia/control/
https://acesectest.adeptia.com/#home

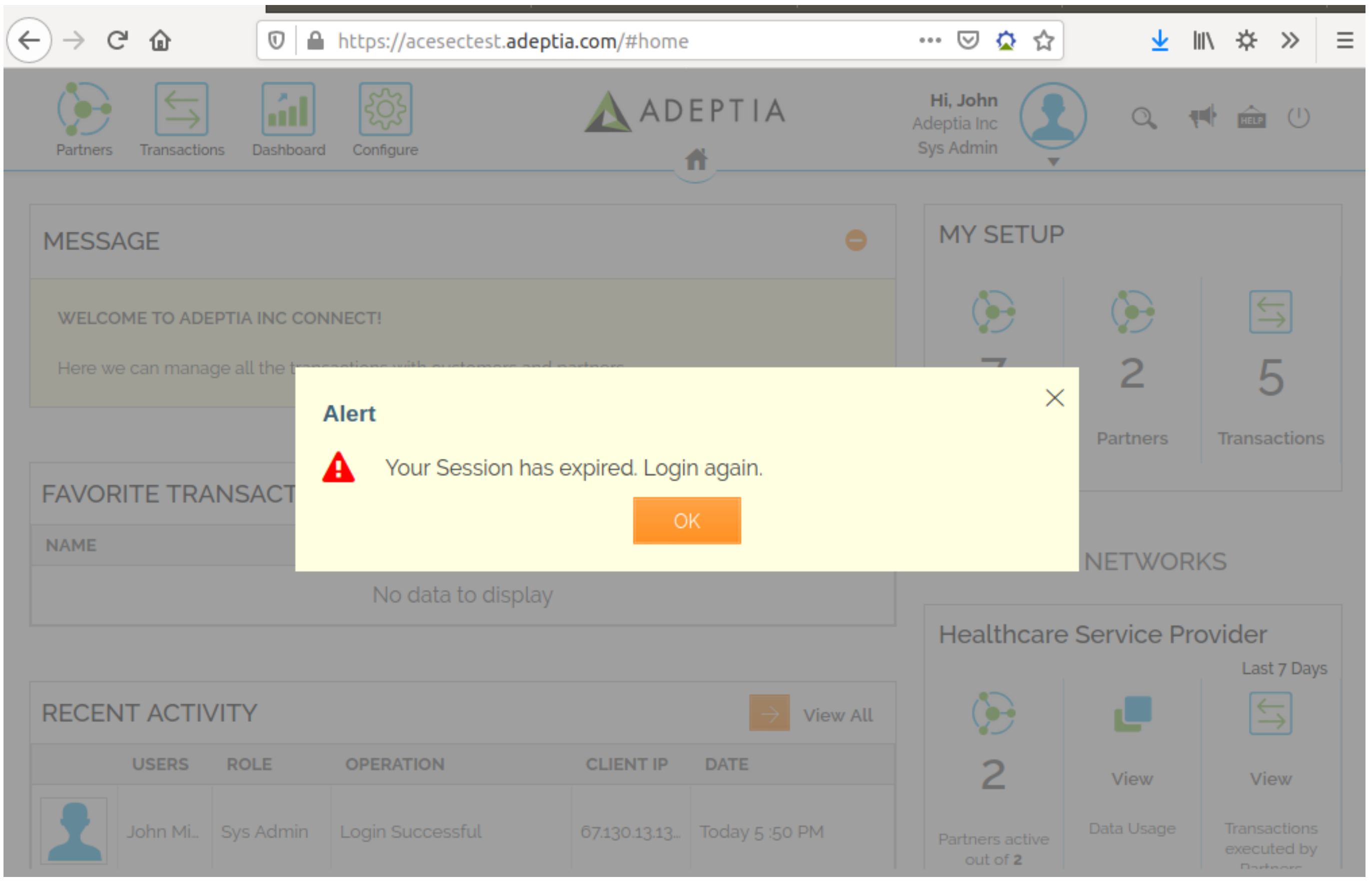
Instance(s):

2

Status:

Remediated

Evidence:



MESSAGE

WELCOME TO ADEPTIA INC CONNECT!

Here we can manage all the transactions with customers and partners

MY SETUP

2 Partners

5 Transactions

Alert

Your Session has expired. Login again.

OK

FAVORITE TRANSACT

NAME
No data to display

NETWORKS

Healthcare Service Provider

Last 7 Days

2 Partners active out of 2

View Data Usage

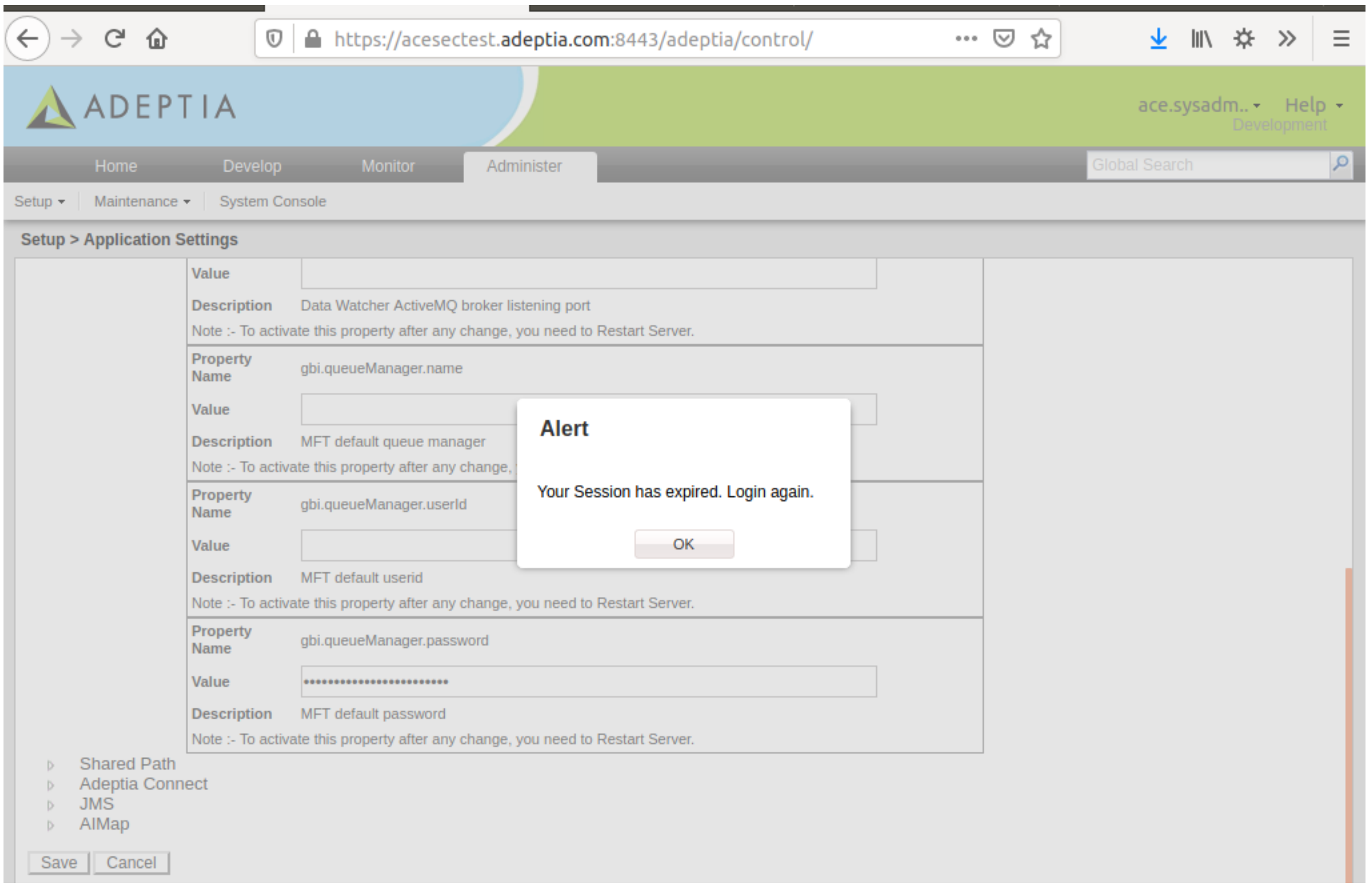
View Transactions executed by Partners

RECENT ACTIVITY

View All

USERS	ROLE	OPERATION	CLIENT IP	DATE
John Mi..	Sys Admin	Login Successful	67.130.13.13..	Today 5 :50 PM

Evidence notes:
Information remains visible on the screen after the session times out, leaving it visible to a passerby.



Setup > Application Settings

Value	
Description	Data Watcher ActiveMQ broker listening port
Note :- To activate this property after any change, you need to Restart Server.	
Property Name	gbi.queueManager.name
Value	
Description	MFT default queue manager
Note :- To activate this property after any change,	
Property Name	gbi.queueManager.userId
Value	
Description	MFT default userid
Note :- To activate this property after any change, you need to Restart Server.	
Property Name	gbi.queueManager.password
Value	************************
Description	MFT default password
Note :- To activate this property after any change, you need to Restart Server.	

Alert

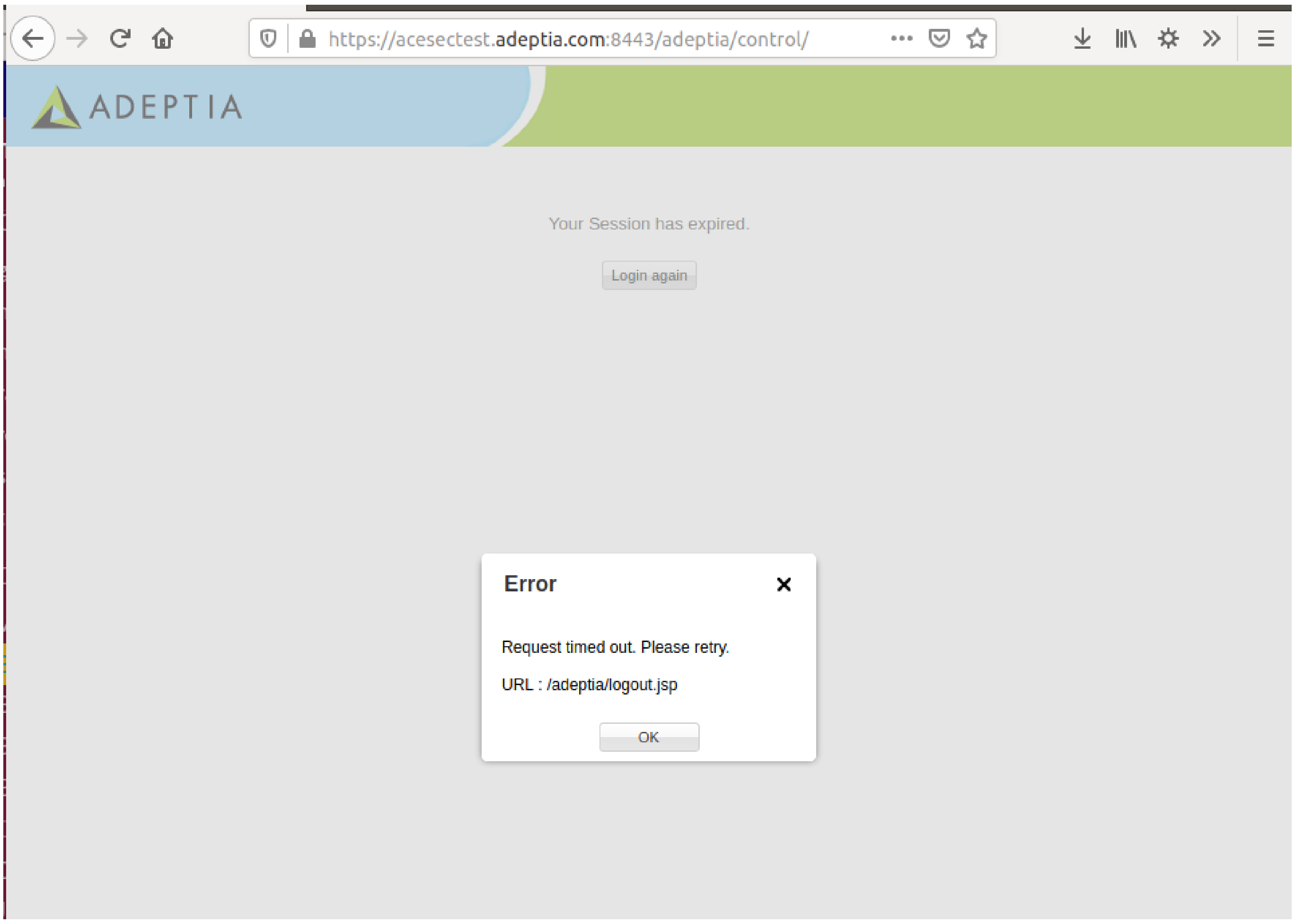
Your Session has expired. Login again.

OK

Shared Path
Adeptia Connect
JMS
AIMap

Save Cancel

Evidence notes:
Image shows information visible in the admin panel after the session times out.



Your Session has expired.

Error

Request timed out. Please retry.

URL : /adeptia/logout.jsp

Evidence notes:
Image shows screen after time out.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2* + ***Impact****(Technical Impact + Business Impact) /2* = ***Risk Rating****(Likelihood + Impact) /2*

Low (3) = (Likelihood (3 + 3) /2 = Low (3) + Impact (3 + 3) /2 = Low (3)) /2

Reference(s):

http://projects.webappsec.org/w/page/13246936/Information Leakage

CVSS:

(AV:N/AC:H/Au:M/C:P/I:P/A:N)

[Back to Top]

[THIS FINDING HAS BEEN REMEDIATED]

Finding(s)

9. Sensitive Information Passed Using GET | Low (2)

Description:

HTTP uses multiple request methods to send data. The method “GET” differs from other request methods because any request data is constructed as part of the URL. When sensitive information is passed using a GET request the information is saved in multiple locations (cached), and in plain text.

Impact:

Once being passed in a GET request sensitive information is saved in locations like the browser history, recently visited sites, and others making the information easily accessible to an attacker.

Test(s) Conducted:

While walking through the site functionality each request method is reviewed, along with the data that is sent. When possible, network monitoring tools are used to identify request methods, and the data being passed.

Finding Comments:

In examining the process of authentication, RedTeam observed prior usernames passed in GET requests. Putting the username in a POST body will protect it from third party logs, browser histories (on the device and synced to other devices), and referer headers.

For more information, see:
https://owasp.org/www-community/vulnerabilities/Information_exposure_through_query_strings_in_url

Recommendations:

Modify the HTML form to specify the request method as “POST”. When making requests to the application ensure that no sensitive information is placed within the URL, or as part of the Hypertext REFerence (HREF) attribute.

Affected System(s):

dev-918046.okta.com/login/getimage?username=ace.businessuser@yopmail.com

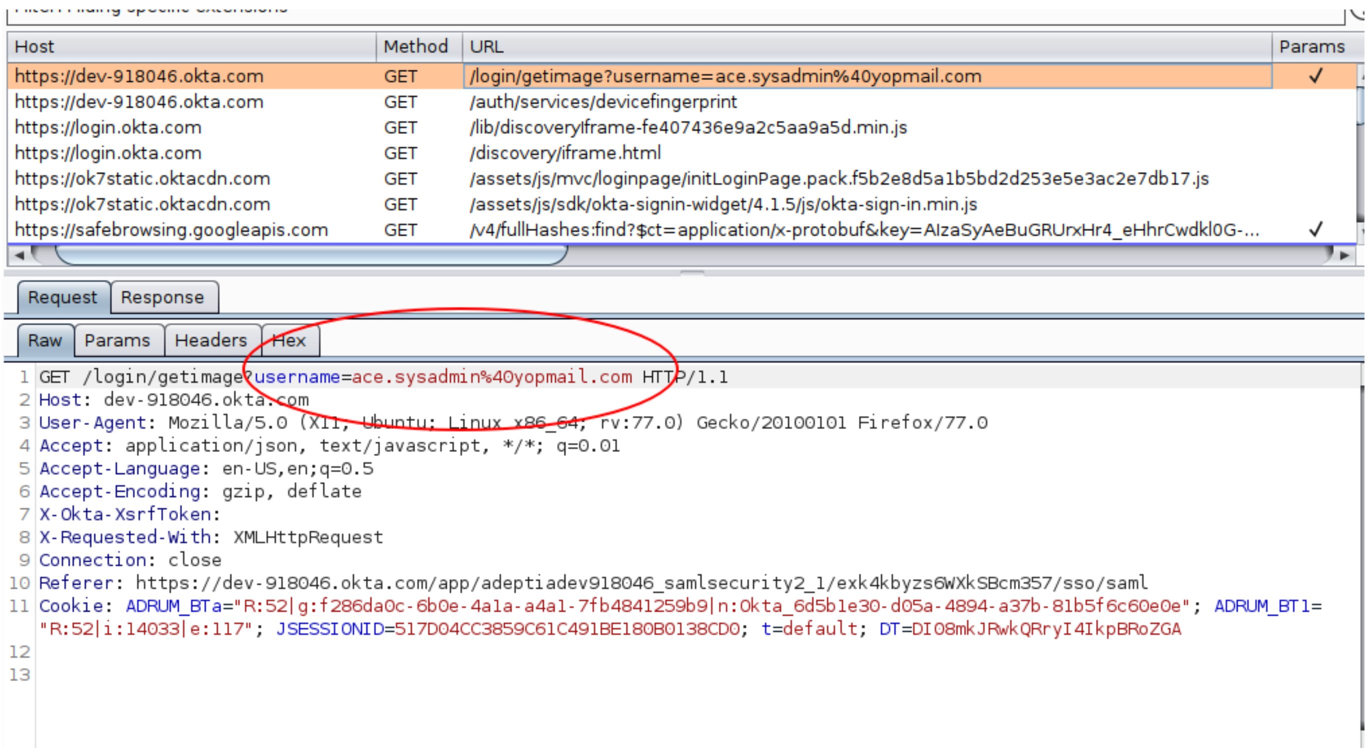
Instance(s):

1

Status:

Not Remediated

Evidence:



Host	Method	URL	Params
https://dev-918046.okta.com	GET	/login/getimage?username=ace.sysadmin%40yopmail.com	✓
https://dev-918046.okta.com	GET	/auth/services/devicefingerprint	
https://login.okta.com	GET	/lib/discoveryIframe-fe407436e9a2c5aa9a5d.min.js	
https://login.okta.com	GET	/discovery/iframe.html	
https://ok7static.oktacdn.com	GET	/assets/js/mvc/loginpage/initLoginPage.pack.f5b2e8d5a1b5bd2d253e5e3ac2e7db17.js	
https://ok7static.oktacdn.com	GET	/assets/js/sdk/okta-signin-widget/4.1.5/js/okta-sign-in.min.js	
https://safebrowsing.googleapis.com	GET	/v4/fullHashes:find?$ct=application/x-protobuf&key=AIzaSyAeBuGRUrxHr4_eHhrCwdkl0G-...	✓

```
GET /login/getimage?username=ace.sysadmin%40yopmail.com HTTP/1.1
Host: dev-918046.okta.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:77.0) Gecko/20100101 Firefox/77.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Okta-XsrfToken:
X-Requested-With: XMLHttpRequest
Connection: close
Referer: https://dev-918046.okta.com/app/adeptiadev918046_samlsecurity2_1/exk4kbyzs6WXkSBcm357/sso/saml
Cookie: ADRUM_BTa="R:52|g:f286da0c-6b0e-4a1a-a4a1-7fb4841259b9|n:Okta_6d5b1e30-d05a-4894-a37b-81b5f6c60e0e"; ADRUM_BT1=
"R:52|i:14033|e:117"; JSESSIONID=517D04CC3859C61C491BE180B0138CD0; t=default; DT=DI08mkJRwkQRryI4IkpBRoZGA
```

Evidence notes:
The username is shown in a GET parameter.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Low (2) = (Likelihood (2 + 2) /2 = Low (2) + Impact (2 + 2) /2 = Low (2)) /2

Reference(s):

https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html
http://www.diffen.com/difference/GET-vs-POST-HTTP-Requests

CVSS:

(AV:A/AC:M/Au:N/C:P/I:P/A:N)

[Back to Top]

Finding(s)

10. Concurrent Login Sessions Allowed | Low (1)

Description:

Users could use one set of authentication credentials to have more than one unique logon session active at the same time. Even though this occurrence may not seem to be related to security it could reduce the granularity of session identifiers or may render logging less useful.

If multiple users can log in to the same account simultaneously, non-repudiation is not possible. This behavior could indicate that a user is attempting to perform malicious activity. Allowing multiple user sessions can allow for a bad actor to control a user’s account at the same time as a legitimate user.

Impact:

This vulnerability could enable higher-risk attacks, such as cloning and hijacking sessions. Also, this can allow a bad actor to access the account at the same time as legitimate users with very little way of identifying the account has been compromised.

Test(s) Conducted:

RedTeam Security logged into the application using the same user account and different browsers. Additional checks are made for alerts or logging that may notify a user that their account is being used in another location or by another device.

Finding Comments:

RedTeam noted that it is possible to have concurrent sessions. This sets up some of the preconditions needed for user session hijack attacks and attacks leveraging stolen credentials if they can occur without detection. If allowing concurrent sessions is determined to have a legitimate business use case, presenting a notice of login activity with IP address would ensure the logged in user is aware of the multiple sessions.

Recommendations:

Consider limiting users to only one session per username. Despite many websites allowing concurrent sessions for the convenience of their users, it presents business risks that must be accepted or addressed. Preventing concurrent sessions can help prevent higher-risk exploits, such as session cloning and session hijacking.

Affected System(s):

acesectest.adeptia.com
acesectest.adeptia.com:8443/adeptia/control/

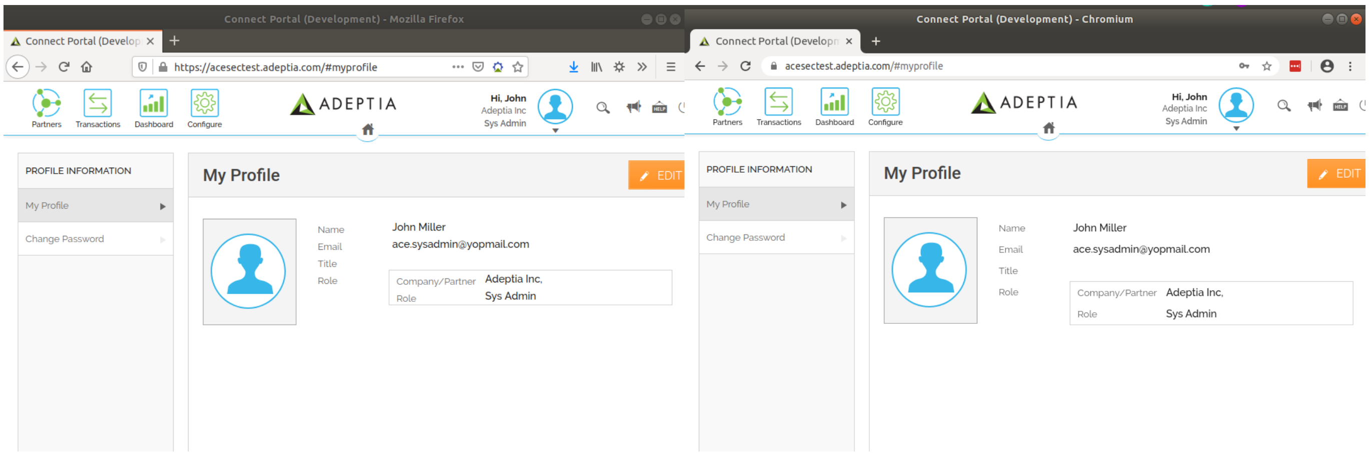
Instance(s):

2

Status:

Not Remediated

Evidence:


Connect Portal (Development) - Mozilla Firefox

https://acesectest.adeptia.com/#myprofile

Partners | Transactions | Dashboard | Configure

ADEPTIA

Hi, John
Adeptia Inc
Sys Admin

PROFILE INFORMATION
My Profile
Change Password

My Profile

EDIT

Name	John Miller
Email	ace.sysadmin@yopmail.com
Title	
Role	
Company/Partner	Adeptia Inc,
Role	Sys Admin

Connect Portal (Development) - Chromium

acesectest.adeptia.com/#myprofile

Partners | Transactions | Dashboard | Configure

ADEPTIA

Hi, John
Adeptia Inc
Sys Admin

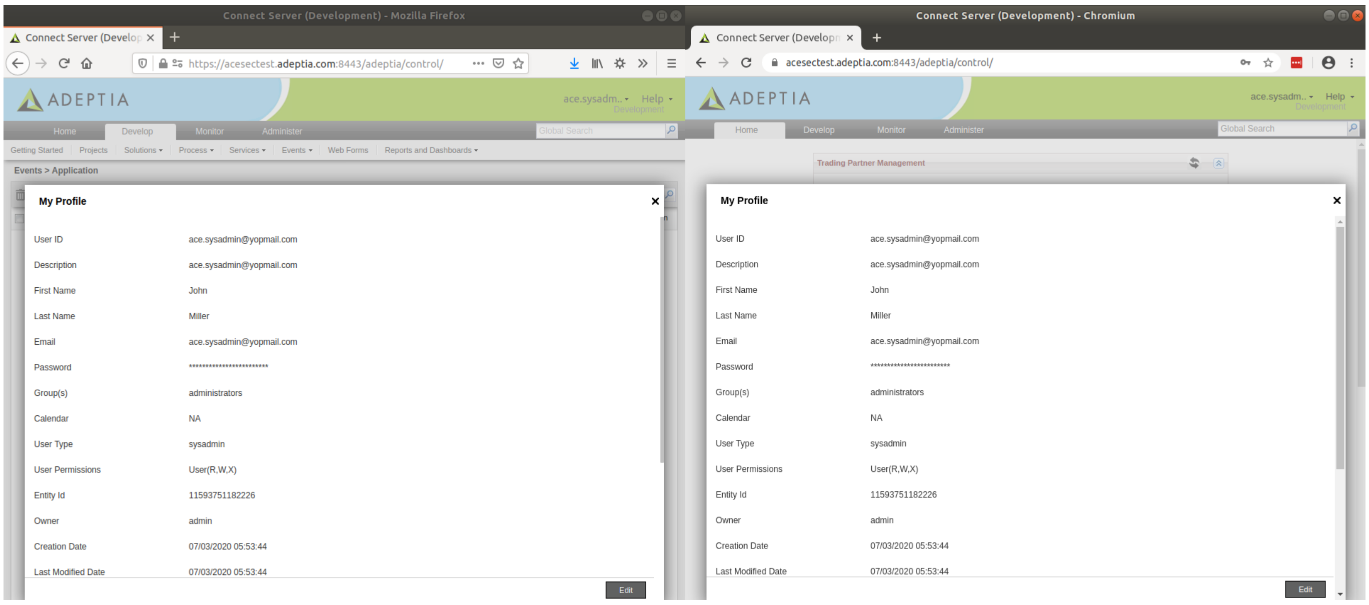
PROFILE INFORMATION
My Profile
Change Password

My Profile

EDIT

Name	John Miller
Email	ace.sysadmin@yopmail.com
Title	
Role	
Company/Partner	Adeptia Inc,
Role	Sys Admin

Evidence notes:
Image shows application with concurrent sessions.


Connect Server (Development) - Mozilla Firefox

https://acesectest.adeptia.com:8443/adeptia/control/

ADEPTIA

ace.sysadm... Help
Development

Home | Develop | Monitor | Administer

Getting Started | Projects | Solutions | Process | Services | Events | Web Forms | Reports and Dashboards

Events > Application

My Profile

User ID	ace.sysadmin@yopmail.com
Description	ace.sysadmin@yopmail.com
First Name	John
Last Name	Miller
Email	ace.sysadmin@yopmail.com
Password	************************
Group(s)	administrators
Calendar	NA
User Type	sysadmin
User Permissions	User(R,W,X)
Entity Id	11593751182226
Owner	admin
Creation Date	07/03/2020 05:53:44
Last Modified Date	07/03/2020 05:53:44

Edit

Connect Server (Development) - Chromium

acesectest.adeptia.com:8443/adeptia/control/

ADEPTIA

ace.sysadm... Help
Development

Home | Develop | Monitor | Administer

Trading Partner Management

My Profile

User ID	ace.sysadmin@yopmail.com
Description	ace.sysadmin@yopmail.com
First Name	John
Last Name	Miller
Email	ace.sysadmin@yopmail.com
Password	************************
Group(s)	administrators
Calendar	NA
User Type	sysadmin
User Permissions	User(R,W,X)
Entity Id	11593751182226
Owner	admin
Creation Date	07/03/2020 05:53:44
Last Modified Date	07/03/2020 05:53:44

Edit

Evidence notes:
Image shows admin application with concurrent sessions.

Severity Calculation:

The process for calculating the finding's severity is derived by assigning a numeric value between 0 and 9 to four (4) criteria separated into Likelihood and Impact. The formula is best represented here: ***Likelihood****(Threat Agents + Vulnerability Factors) /2 +* ***Impact****(Technical Impact + Business Impact) /2 =* ***Risk Rating****(Likelihood + Impact) /2*

Low (1) = (Likelihood (1 + 1) /2 = Low (1) + Impact (1 + 1) /2 = Low (1)) /2

Reference(s):

https://www.owasp.org/index.php/Session_Management_Cheat_Sheet#Simultaneous_Session_Logons

CVSS:

(AV:N/AC:H/Au:M/C:P/I:P/A:P)

[Back to Top]

Appendix A

Approach

RedTeam Security's application penetration test combines the results from industry-leading scanning tools with manual testing to enumerate and validate vulnerabilities, configuration errors, and business logic flaws. In-depth manual application testing enables us to find what scanners often miss.

Web applications are particularly vulnerable to external attack given that they are inherently designed to be accessible to the Internet. While automated scanners check for known vulnerabilities, they are incapable of actually reporting on real business risk. Our web application security testing helps you lower your risk of data breach, improve productivity, protect your brand, and maximize the ROI from your web applications.

RedTeam Security's application penetration test service utilizes a risk-based approach to manually identify critical application-centric vulnerabilities that exist on all in-scope applications.

Using this approach, RedTeam's comprehensive approach covers the classes of vulnerabilities in the Open Web Application Security Project (OWASP) Top 10 2017 and beyond:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging & Monitoring

Automated vs Manual Testing

RedTeam's approach consists of about 80% manual testing and about 20% automated testing - actual results may vary slightly. While automated testing enables efficiency, it is effective in providing efficiency only during the initial phases of a penetration test. At RedTeam Security, it is our belief that an effective and comprehensive test can only be realized through rigorous manual testing techniques.

Tools

In order to perform a comprehensive real-world assessment, RedTeam Security utilizes commercial tools, internally developed tools and the same tools that hacker use on each and every assessment. Once again, our intent is to assess systems by simulating a real-world attack and we leverage the many tools at our disposal to effectively carry out that task.

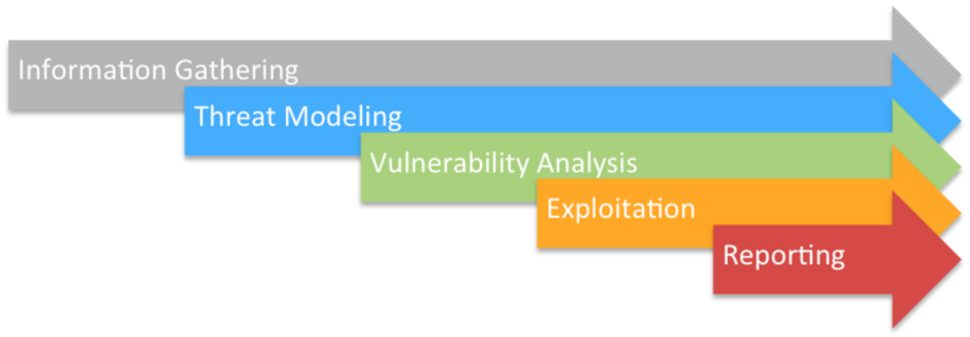
We make use of tools from the following categories (not a complete list):

- Commercial tools (i.e.: Burp Suite Pro, AppScan, WebInspect)
- Open source / Hacker tools (i.e.: Metasploit, BEeF, Kali Linux, OWASP Zap)
- RedTeam developed tools (i.e.: nmapcli, Metasploit modules, PlugBot, various scripts)

Appendix A

Methodology

Penetration Testing Methodology



Information Gathering

The information-gathering phase consists of Google search engine reconnaissance, server fingerprinting, application enumeration and more. Information gathering efforts results in a compiled list of metadata and raw output with the goal of obtaining as much information about the application's makeup as possible. Reconnaissance includes initial domain foot printing, metafile leakage review, service enumeration and operating system and application fingerprinting. The purpose of this step is to collectively map the in-scope environment and prepare for threat identification.
During this phase, RedTeam Security will perform the following:

- Use discovery tools to passively uncover information about the application (ie: robots.txt)
- Identify entry points into the application, such as administration portals or backdoors
- Perform application fingerprinting, in order to identify the use of a CMS (ie: Drupal) and the underlying dev language
- Send fuzzing requests to be used in the analysis of error codes that may disclose valuable information that could be used to launch a more targeted attack
- Actively scan for open services and develop a test plan for latter phases in the assessment

Threat Modeling

With the information collected from the previous step, security testing transitions to identifying vulnerabilities in the application. This typically begins with automated scans (i.e.: AppScan) initially but quickly morphs into manual testing techniques using more pointed and direct tools. During the threat-modeling step, assets are identified and categorized into threat categories. These may involve: sensitive documents, trade secrets, financial information, etc.
During this phase, RedTeam Security will perform the following:

- Use open source, commercial and internally developed tools to identify well-known vulnerabilities (ie: AppScan, BURP, WebInspect, Metasploit)
- Spider the in-scope application(s) to effectively build a map of each of the features, components and areas of interest
- Use discovered sections, features, capabilities to establish threat categories to be used for more manual/rigorous testing (ie: file uploads, admin backdoors, web services, WYSIWYG editors)

Appendix A

- Send fuzzing requests to be used in the analysis of error codes that may disclose valuable information that could be used to launch a more targeted attack
- Build the application's threat model using the information gathered in this phase. This model will be used as a plan of attack for latter phases in the assessment.

Vulnerability Analysis

The vulnerability analysis step involves the documenting and analysis of vulnerabilities discovered as a result of the previous steps. This includes the analysis of out from the various security tools and manual testing techniques.
During this phase, RedTeam Security will perform the following:

- Compile the list of areas of interest and develop a plan for exploitation
- Search and gather known exploits from various sources (ExploitDB, Pastebin, etc)
- Analyze the impact and likelihood for each potential exploitable vulnerability
- Select the best method and tools for properly exploiting each of the suspected exploitable vulnerabilities

Exploitation

Unlike a vulnerability assessment, a penetration test takes such a test quite a bit further by way of exploitation. Exploitation involves establishing access to application through the bypassing and exploitation of security controls in order to determine their actual real world risk. Throughout this step, we perform several manual tests incapable of being performed through automated means, such as scanners. During a RedTeam Security penetration test, this phase consists of heavy manual testing tactics and is often the most time-intensive phase. Exploitation may include, but is not limited to: buffer overflow, SQL injection, OS commanding, cross-site scripting and more.

During this phase, RedTeam Security will perform the following:

- Using the identified vulnerabilities in the previous phase, RedTeam will manually exploit any identified vulnerabilities in order to validate them
- Capture and log evidence to provide proof of exploitation (ie: images, movies, screenshots, configs, etc.)
- Notify the client of any Critical or High findings upon discovery by telephone and email
- Upload validated exploits and their corresponding evidence/information to the project portal for client review
- Perform re-testing, per client request

Reporting

The reporting step is intended to deliver, rank and prioritize findings and generate a clear and actionable report, complete with evidence, to the project stakeholders. The presentation of findings can occur via Webex or in-person - whichever format is most conducive for communicating results.

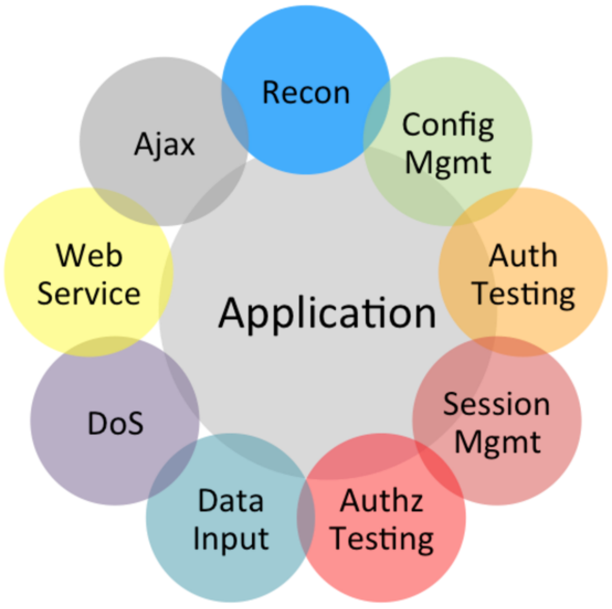
Appendix A

During this phase, RedTeam Security will perform the following:

- Ensure all findings have been uploaded to the project portal for client review
- Create the penetration test report, along with evidence, and upload it to the client portal for review
- Schedule a meeting with the client in an effort to present and talk through each of the identified vulnerabilities
- Optionally, additional meeting may take place to ensure the client understands the findings and recommendations for mitigation

Comprehensive Methodology

Each and every internal penetration test is conducted consistently using globally accepted and industry standard frameworks. In order to ensure a sound and comprehensive penetration test, RedTeam leverages industry standard frameworks as a foundation for carrying out penetration tests. The underlying framework is based on the Open Web Application Security Project (OWASP).



OWASP is a globally accepted framework designed to enable the execution of effective penetration testing consistent with best practice all while ensuring a holistic and comprehensive evaluation. At RedTeam Security, we consider this phase to be the most important and we take great care to ensure we've communicated the value of our service and findings thoroughly.

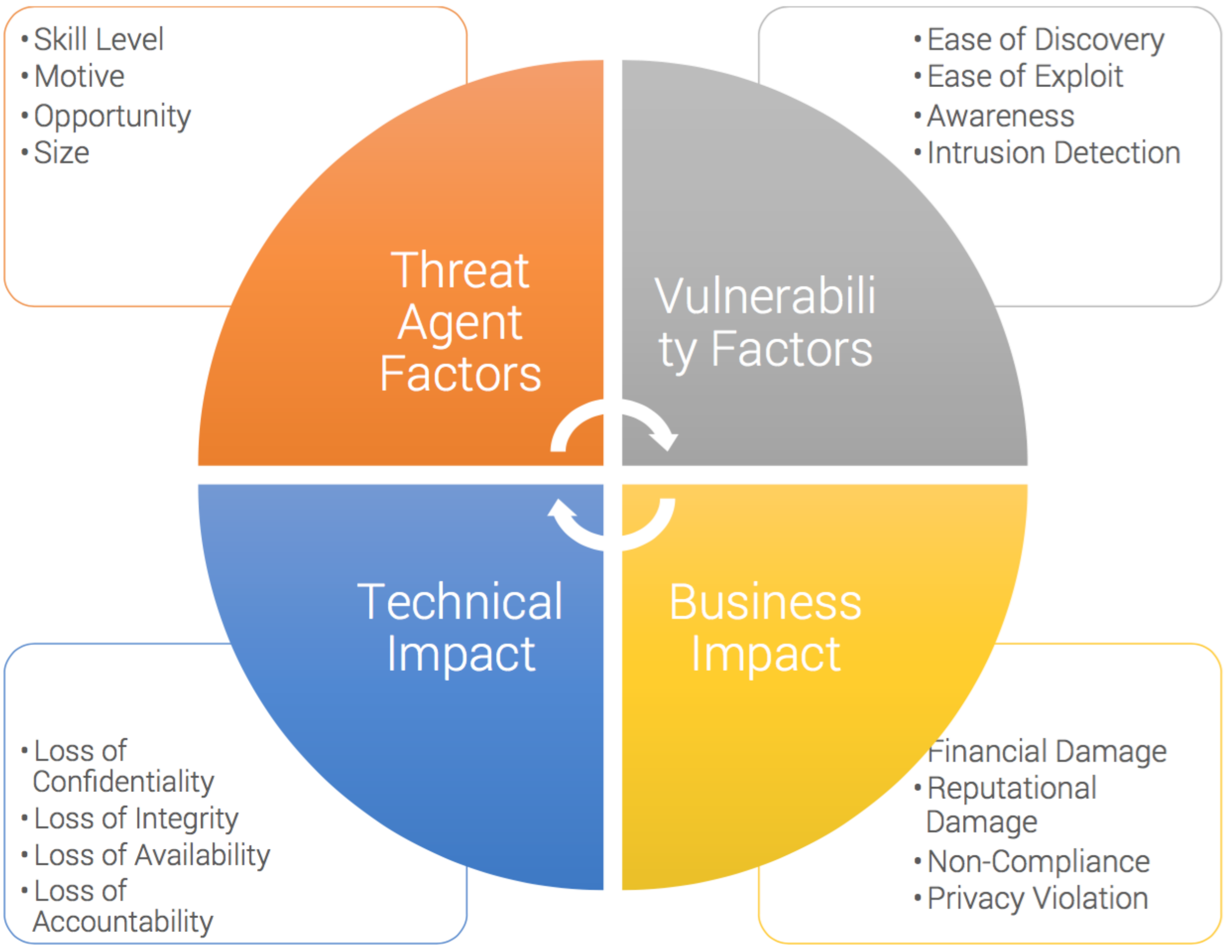
Appendix B

Risk Rating Overview

RedTeam Security has adopted an industry-standard approach to assigning risk ratings to vulnerabilities. This approach is used in all our assessments and provides our clients with risk ratings that take into account a number of factors ranging from: Skill Level, Motive, Ease of Exploit, Loss of Integrity to Privacy/Reputational Damage.

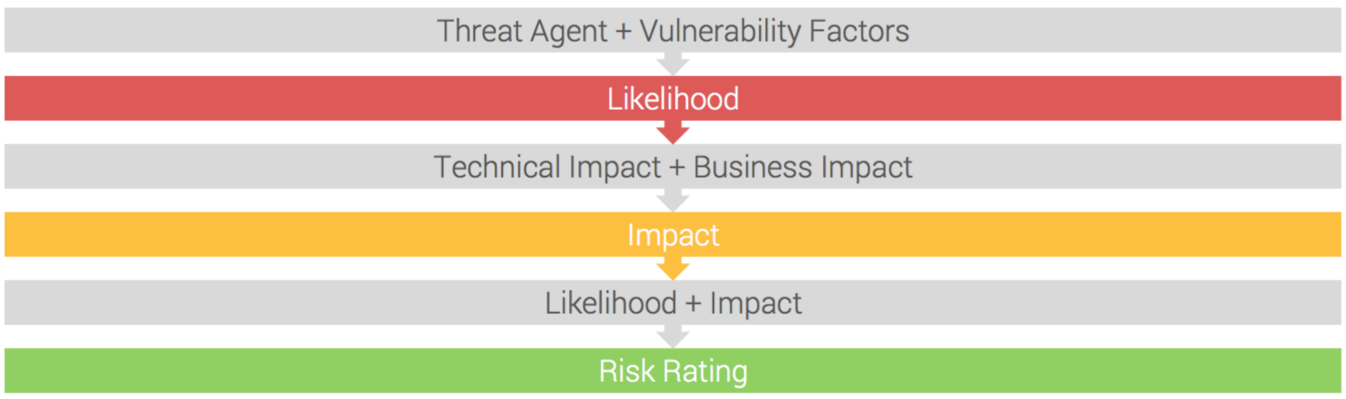
Our comprehensive approach ensures that our clients' vulnerabilities are represented by their true real-world likelihood and potential impact to their business.

Risk Rating Factors



Appendix B

Risk Calculation


Threat Agent + Vulnerability Factors
Likelihood
Technical Impact + Business Impact
Impact
Likelihood + Impact
Risk Rating

Risk Calculation is carried out through a quantitative method. The calculation is an industry standard approach and is widely adopted by many organizations across the globe. Please see the detail below for a walkthrough of the risk calculation process.

Calculation of *Likelihood* is achieved by the equation:

AVERAGE(Threat Agent + Vulnerability Factors) = Likelihood

Calculation of *Impact* is achieved by the equation:

AVERAGE(Technical Impact + Business Impact) = Impact

Calculation of the finding's overall *Risk Rating* is achieved by the following equation:

AVERAGE(Likelihood + Impact) = Risk Rating

Factors Explained

THREAT AGENT FACTORS

Factors in this category aid in establishing the real-world likelihood of exploitation. Overall, these factors take into account the knowledge and breadth of the threat.

- Skill Level – How technically skilled are the group of agents
- Motive – How motivated are the group of agents
- Opportunity – What resources/opportunity are required to find/exploit
- Size – How large is the group of agents

Appendix B

VULNERABILITY FACTORS

Factors in this category aid in establishing the real-world likelihood of exploitation. Overall, these factors take into account the ease of exploitation and how well known it might be.

- Ease of Discovery – How easy is it to discover this vulnerability
- Ease of Exploit – How easy is it to actually exploit this vulnerability
- Awareness – How well known is this vulnerability
- Intrusion Detection – How likely is this to be exploited

TECHNICAL IMPACT FACTORS

Factors in this category aid in establishing the estimated impact. Overall, these factors account for potential damage to CIA (Confidentiality, Integrity, Availability) with respect to data.

- Loss of Confidentiality – How much data could be disclosed and how sensitive
- Loss of Integrity – How much data could be corrupted/damaged
- Loss of Availability – How much service could be lost and how vital is it
- Loss of Accountability – Are the threat agents' action traceable to an individual

BUSINESS IMPACT FACTORS

Factors in this category aid in establishing the estimated impact. Overall, these factors account for potential damage to the business, such as reputation, finances and privacy.

- Financial Damage – How much financial damage would result
- Reputational Damage – Would an exploit cause reputational damage
- Non-Compliance – How much does exposure does non-compliance introduce
- Privacy Violation – How much personally identifiable information could be disclosed

Appendix C

Tools

Shown below is a list of the most commonly used tools during such an engagement. RedTeam Security consultants utilize commercial, open source and RedTeam-developed tools. Be advised this is not an completed and exhaustive list.

Nessus	nmap
Kali Linux	Wireshark
Metasploit	nmapcli
PlugBot	John the Ripper
Hydra	Nikto
OpenVAS	Cain & Abel
Olly Debugger	IDA Pro
hping	onesixtyone
AppScan	WebInspect
Hydra	Burp Suite Pro
Firewalk	fragroute / fragrouter
sqlmap	netifera
sslscan	Forify SCA
scapy	Mantra
TOR	Ethereal
sslscan	Forify SCA
i2p	tcpdump
OWASP ZAP	Aircrack
BeEF Framework	OWASP Xenotix XSS Exploit Framework
Spike	Cookiedigger
Paros Proxy	dsniff
Brutus	P0f
Kismet	dnsenum
Maltego	Skipfish
Social Engineering Toolkit	Armitage