



Sign in

Kevin-Robertson / Inveigh Public

Notifications

Fork 444

Star 2.5k

<> Code

Issues 19

Pull requests 1

Actions

Projects

Wiki

Security

Insights

Inveigh / Inveigh / Support / Output.cs



1570 lines (1287 loc) · 57.9 KB

Code

Blame

Raw



```
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Threading;
5  using System.Linq;
6  using System.Diagnostics;
7  using Quiddity.Support;
8
9  namespace Inveigh
10 {
11     class Output
12     {
13
14         public static void OutputLoop()
15         {
16             bool keyDetect = true;
17             bool keyPressed = false;
18
19             do
20             {
21
22                 while (Program.enabledConsoleOutput && !keyPressed)
23                 {
24
25                     try
26                     {
```

```
27
28         if (keyDetect && Console.KeyAvailable)
29         {
30             keyPressed = true;
31         }
32
33     }
34     catch { keyDetect = false; }
35
36     while (Program.consoleList.Count > 0)
37     {
38         ConsoleOutputFormat(Program.consoleList[0]);
39         Program.consoleList.RemoveAt(0);
40     }
41
42     if (!Program.isRunning)
43     {
44         break;
45     }
46
47     Thread.Sleep(5);
48 }
49
50 } while (Program.isRunning && Program.enabledConsoleOutput && Console.ReadKey(true).Key
51
52 }
53
54 ✓ public static void Queue(string Output)
55 {
56
57     lock (Program.outputList)
58     {
59         Program.outputList.Add(Output);
60     }
61
62 }
63
64 ✓ public static void OutputColor(string output, string status, ConsoleColor color)
65 {
66     string[] split = output.Substring(1).Split(' ');
67
68     foreach (string segment in split)
69     {
70         string[] split2 = segment.Split('');
71
72         int i = 0;
```

```
73         foreach (string segment2 in split2)
74         {
75             int j = 0;
76             if (i % 2 == 0)
77             {
78                 string[] split3 = segment2.Split('|');
79                 Console.Write("[");
80
81                 foreach (string segment3 in split3)
82                 {
83
84                     if (j != 0 && j < split3.Length)
85                     {
86                         Console.Write("|");
87                     }
88
89                     Console.ForegroundColor = color;
90                     Console.Write(segment3);
91                     Console.ResetColor();
92                     j++;
93                 }
94
95                 Console.Write("]");
96             }
97             else
98             {
99
100                 if (segment2.Contains("\r\n"))
101                 {
102                     string[] split4 = segment2.Split('\n');
103
104                     if (split4.Length == 2)
105                     {
106                         Console.Write(split4[0] + "\n");
107                         Console.ForegroundColor = color;
108                         Console.Write(split4[1]);
109                         Console.ResetColor();
110                     }
111                     else
112                     {
113                         Console.Write(segment2);
114                     }
115
116                 }
117             else
118             {
```

110

1



































































```

1497     }
1498
1499     while (Program.ntlmv1FileList.Count > 0)
1500     {
1501
1502         using (StreamWriter outputFileNTLMv1 = new StreamWriter(Path.Combine(Program.argFil
1503         {
1504             outputFileNTLMv1.WriteLine(Program.ntlmv1FileList[0]);
1505             outputFileNTLMv1.Close();
1506
1507             lock (Program.ntlmv1FileList)
1508             {
1509                 Program.ntlmv1FileList.RemoveAt(0);
1510             }
1511
1512         }
1513
1514     }
1515
1516     while (Program.ntlmv2FileList.Count > 0)
1517     {
1518
1519         using (StreamWriter outputFileNTLMv2 = new StreamWriter(Path.Combine(Program.argFil
1520         {
1521             outputFileNTLMv2.WriteLine(Program.ntlmv2FileList[0]);
1522             outputFileNTLMv2.Close();
1523
1524             lock (Program.ntlmv2FileList)
1525             {
1526                 Program.ntlmv2FileList.RemoveAt(0);
1527             }
1528
1529         }
1530
1531     }
1532
1533     while (Program.ntlmv1UsernameFileList.Count > 0)
1534     {
1535
1536         using (StreamWriter outputFileNTLMv1 = new StreamWriter(Path.Combine(Progra
1537         {
1538             outputFileNTLMv1.WriteLine(Program.ntlmv1UsernameFileList[0]);
1539             outputFileNTLMv1.Close();
1540
1541             lock (Program.ntlmv1UsernameFileList)
1542             {
1543                 Program.ntlmv1UsernameFileList.RemoveAt(0);
1544             }
1545
1546         }
1547
1548     }
1549
1550     while (Program.ntlmv2UsernameFileList.Count > 0)
1551     {
1552         using (StreamWriter outputFileNTLMv2 = new StreamWriter(Path.Combine(Progra
1553         {
1554             outputFileNTLMv2.WriteLine(Program.ntlmv2UsernameFileList[0]);
1555             outputFileNTLMv2.Close();
1556
1557             lock (Program.ntlmv2UsernameFileList)
1558             {
1559                 Program.ntlmv2UsernameFileList.RemoveAt(0);
1560             }
1561
1562         }
1563
1564     }
1565
1566     Console.WriteLine("NTLMv1 and NTLMv2 hashes and usernames saved to files.");
1567     Console.WriteLine("Press any key to exit.");
1568     Console.ReadKey();
1569 }
1570
1571 // Program exit
1572 return 0;
1573 }
1574
1575 // Program exit
1576 return 0;
1577 }
1578
1579 // Program exit
1580 return 0;
1581 }
1582
1583 // Program exit
1584 return 0;
1585 }
1586
1587 // Program exit
1588 return 0;
1589 }
1590
1591 // Program exit
1592 return 0;
1593 }
1594
1595 // Program exit
1596 return 0;
1597 }
1598
1599 // Program exit
1600 return 0;
1601 }
1602
1603 // Program exit
1604 return 0;
1605 }
1606
1607 // Program exit
1608 return 0;
1609 }
1610
1611 // Program exit
1612 return 0;
1613 }
1614
1615 // Program exit
1616 return 0;
1617 }
1618
1619 // Program exit
1620 return 0;
1621 }
1622
1623 // Program exit
1624 return 0;
1625 }
1626
1627 // Program exit
1628 return 0;
1629 }
1630
1631 // Program exit
1632 return 0;
1633 }
1634
1635 // Program exit
1636 return 0;
1637 }
1638
1639 // Program exit
1640 return 0;
1641 }
1642
1643 // Program exit
1644 return 0;
1645 }
1646
1647 // Program exit
1648 return 0;
1649 }
1650
1651 // Program exit
1652 return 0;
1653 }
1654
1655 // Program exit
1656 return 0;
1657 }
1658
1659 // Program exit
1660 return 0;
1661 }
1662
1663 // Program exit
1664 return 0;
1665 }
1666
1667 // Program exit
1668 return 0;
1669 }
1670
1671 // Program exit
1672 return 0;
1673 }
1674
1675 // Program exit
1676 return 0;
1677 }
1678
1679 // Program exit
1680 return 0;
1681 }
1682
1683 // Program exit
1684 return 0;
1685 }
1686
1687 // Program exit
1688 return 0;
1689 }
1690
1691 // Program exit
1692 return 0;
1693 }
1694
1695 // Program exit
1696 return 0;
1697 }
1698
1699 // Program exit
1700 return 0;
1701 }
1702
1703 // Program exit
1704 return 0;
1705 }
1706
1707 // Program exit
1708 return 0;
1709 }
1710
1711 // Program exit
1712 return 0;
1713 }
1714
1715 // Program exit
1716 return 0;
1717 }
1718
1719 // Program exit
1720 return 0;
1721 }
1722
1723 // Program exit
1724 return 0;
1725 }
1726
1727 // Program exit
1728 return 0;
1729 }
1730
1731 // Program exit
1732 return 0;
1733 }
1734
1735 // Program exit
1736 return 0;
1737 }
1738
1739 // Program exit
1740 return 0;
1741 }
1742
1743 // Program exit
1744 return 0;
1745 }
1746
1747 // Program exit
1748 return 0;
1749 }
1750
1751 // Program exit
1752 return 0;
1753 }
1754
1755 // Program exit
1756 return 0;
1757 }
1758
1759 // Program exit
1760 return 0;
1761 }
1762
1763 // Program exit
1764 return 0;
1765 }
1766
1767 // Program exit
1768 return 0;
1769 }
1770
1771 // Program exit
1772 return 0;
1773 }
1774
1775 // Program exit
1776 return 0;
1777 }
1778
1779 // Program exit
1780 return 0;
1781 }
1782
1783 // Program exit
1784 return 0;
1785 }
1786
1787 // Program exit
1788 return 0;
1789 }
1790
1791 // Program exit
1792 return 0;
1793 }
1794
1795 // Program exit
1796 return 0;
1797 }
1798
1799 // Program exit
1800 return 0;
1801 }
1802
1803 // Program exit
1804 return 0;
1805 }
1806
1807 // Program exit
1808 return 0;
1809 }
1810
1811 // Program exit
1812 return 0;
1813 }
1814
1815 // Program exit
1816 return 0;
1817 }
1818
1819 // Program exit
1820 return 0;
1821 }
1822
1823 // Program exit
1824 return 0;
1825 }
1826
1827 // Program exit
1828 return 0;
1829 }
1830
1831 // Program exit
1832 return 0;
1833 }
1834
1835 // Program exit
1836 return 0;
1837 }
1838
1839 // Program exit
1840 return 0;
1841 }
1842
1843 // Program exit
1844 return 0;
1845 }
1846
1847 // Program exit
1848 return 0;
1849 }
1850
1851 // Program exit
1852 return 0;
1853 }
1854
1855 // Program exit
1856 return 0;
1857 }
1858
1859 // Program exit
1860 return 0;
1861 }
1862
1863 // Program exit
1864 return 0;
1865 }
1866
1867 // Program exit
1868 return 0;
1869 }
1870
1871 // Program exit
1872 return 0;
1873 }
1874
1875 // Program exit
1876 return 0;
1877 }
1878
1879 // Program exit
1880 return 0;
1881 }
1882
1883 // Program exit
1884 return 0;
1885 }
1886
1887 // Program exit
1888 return 0;
1889 }
1890
1891 // Program exit
1892 return 0;
1893 }
1894
1895 // Program exit
1896 return 0;
1897 }
1898
1899 // Program exit
1900 return 0;
1901 }
1902
1903 // Program exit
1904 return 0;
1905 }
1906
1907 // Program exit
1908 return 0;
1909 }
1910
1911 // Program exit
1912 return 0;
1913 }
1914
1915 // Program exit
1916 return 0;
1917 }
1918
1919 // Program exit
1920 return 0;
1921 }
1922
1923 // Program exit
1924 return 0;
1925 }
1926
1927 // Program exit
1928 return 0;
1929 }
1930
1931 // Program exit
1932 return 0;
1933 }
1934
1935 // Program exit
1936 return 0;
1937 }
1938
1939 // Program exit
1940 return 0;
1941 }
1942
1943 // Program exit
1944 return 0;
1945 }
1946
1947 // Program exit
1948 return 0;
1949 }
1950
1951 // Program exit
1952 return 0;
1953 }
1954
1955 // Program exit
1956 return 0;
1957 }
1958
1959 // Program exit
1960 return 0;
1961 }
1962
1963 // Program exit
1964 return 0;
1965 }
1966
1967 // Program exit
1968 return 0;
1969 }
1970
1971 // Program exit
1972 return 0;
1973 }
1974
1975 // Program exit
1976 return 0;
1977 }
1978
1979 // Program exit
1980 return 0;
1981 }
1982
1983 // Program exit
1984 return 0;
1985 }
1986
1987 // Program exit
1988 return 0;
1989 }
1990
1991 // Program exit
1992 return 0;
1993 }
1994
1995 // Program exit
1996 return 0;
1997 }
1998
1999 // Program exit
2000 return 0;
2001 }
2002
2003 // Program exit
2004 return 0;
2005 }
2006
2007 // Program exit
2008 return 0;
2009 }
2010
2011 // Program exit
2012 return 0;
2013 }
2014
2015 // Program exit
2016 return 0;
2017 }
2018
2019 // Program exit
2020 return 0;
2021 }
2022
2023 // Program exit
2024 return 0;
2025 }
2026
2027 // Program exit
2028 return 0;
2029 }
2030
2031 // Program exit
2032 return 0;
2033 }
2034
2035 // Program exit
2036 return 0;
2037 }
2038
2039 // Program exit
2040 return 0;
2041 }
204
```

```
1538             outputUsernameFileNTLMv1.WriteLine(Program.ntlmv1UsernameFileList[0]);
1539             outputUsernameFileNTLMv1.Close();
1540
1541             lock (Program.ntlmv1UsernameList)
1542             {
1543                 Program.ntlmv1UsernameFileList.RemoveAt(0);
1544             }
1545
1546         }
1547
1548     }
1549
1550     while (Program.ntlmv2UsernameFileList.Count > 0)
1551     {
1552
1553         using (StreamWriter outputUsernameFileNTLMv2 = new StreamWriter(Path.Combine(Program.
1554         {
1555             outputUsernameFileNTLMv2.WriteLine(Program.ntlmv2UsernameFileList[0]);
1556             outputUsernameFileNTLMv2.Close();
1557
1558             lock (Program.ntlmv2UsernameFileList)
1559             {
1560                 Program.ntlmv2UsernameFileList.RemoveAt(0);
1561             }
1562
1563         }
1564
1565     }
1566
1567 }
1568
1569 }
1570 }
```