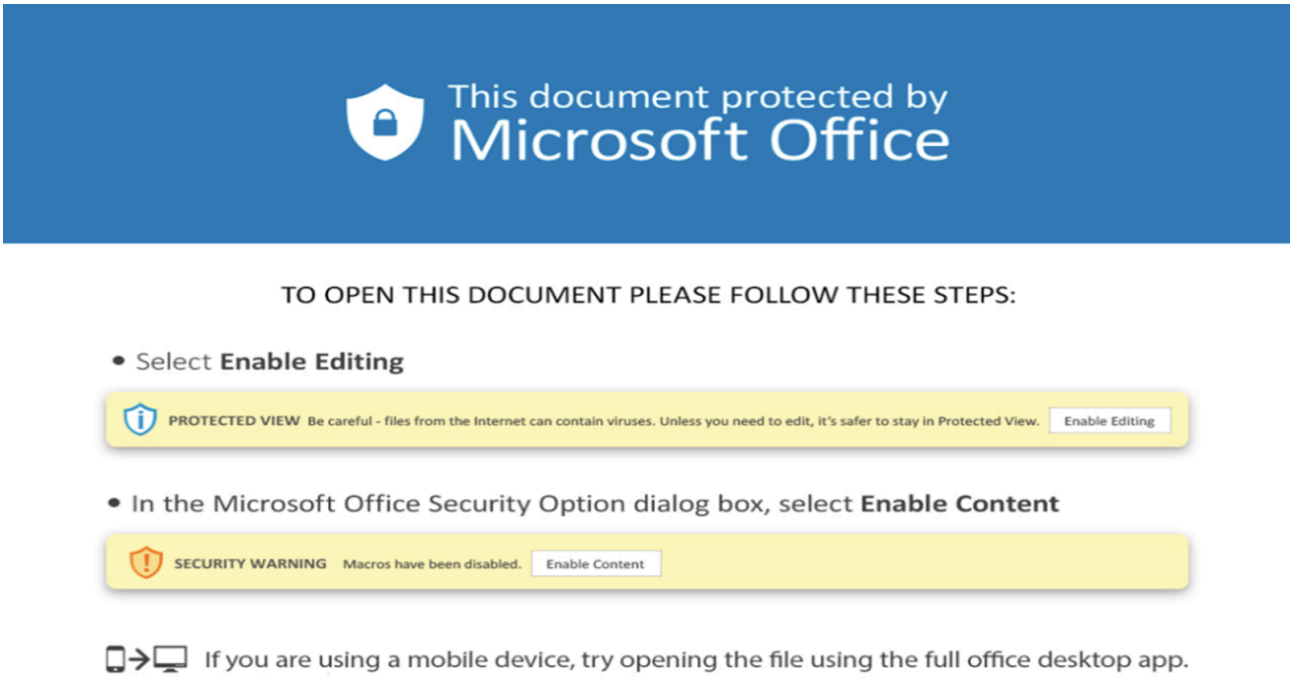
XLSB Files: Because Binary is Stealthier Than XML - SANS ISC

By SANS Internet Storm Center

Archived: 2026-04-05 14:49:00 UTC

In one of his last diaries[1], Brad mentioned an Excel sheet named with a .xlsb extension. Now, it was my turn to find one... What's the magic behind this file extension? "XLS" means that we are facing an Excel sheet and "B" means that we have a binary workbook file. Within the current Microsoft office files format, data are stored in XML. In this case, they are stored in binary. For Microsoft Office, to open a normal or binary file is the same... but for an attacker, the plus-value is the increased level of obfuscation! Indeed, it's more difficult to extract interesting information like... strings!

When you open the file, you see a classic message asking you to enable macros:



Let's have a look at the document:

```
remnux@remnux:/MalwareZoo/20220324$ oledump.py luamccsbuaraeos.xlsb
Warning: no OLE file was found inside this ZIP container (OPC)
```

No VBA macro found! Let's check the content of the file and look for a potential Excel 4 macro:

```
remnux@remnux:/mnt/hgfs/MalwareZoo/20220324$ zipdump.py luamccsbuaraeos.xlsb
Index Filename                                  Encrypted Timestamp
    1 [Content_Types].xml                               0 1980-01-01 00:00:00
```

```
    2 _rels/.rels                                       0 1980-01-01 00:00:00
    3 xl/_rels/workbook.bin.rels                        0 1980-01-01 00:00:00
    4 xl/workbook.bin                                   0 1980-01-01 00:00:00
    5 xl/worksheets/sheet1.bin                          0 1980-01-01 00:00:00
    6 xl/worksheets/sheet2.bin                          0 1980-01-01 00:00:00
    7 xl/worksheets/sheet3.bin                          0 1980-01-01 00:00:00
    8 xl/worksheets/sheet4.bin                          0 1980-01-01 00:00:00
    9 xl/macrosheets/intlsheet1.bin                     0 1980-01-01 00:00:00
   10 xl/macrosheets/sheet1.bin                         0 1980-01-01 00:00:00
   11 xl/macrosheets/sheet2.bin                         0 1980-01-01 00:00:00
   12 xl/theme/theme1.xml                               0 1980-01-01 00:00:00
   13 xl/media/image1.png                               0 1980-01-01 00:00:00
   14 xl/styles.bin                                     0 1980-01-01 00:00:00
   15 xl/drawings/drawing1.xml                          0 1980-01-01 00:00:00
   16 xl/worksheets/_rels/sheet1.bin.rels               0 1980-01-01 00:00:00
   17 xl/worksheets/_rels/sheet2.bin.rels               0 1980-01-01 00:00:00
   18 xl/worksheets/_rels/sheet3.bin.rels               0 1980-01-01 00:00:00
   19 xl/worksheets/_rels/sheet4.bin.rels               0 1980-01-01 00:00:00
   20 xl/macrosheets/_rels/intlsheet1.bin.rels          0 1980-01-01 00:00:00
   21 xl/macrosheets/_rels/sheet1.bin.rels              0 1980-01-01 00:00:00
   22 xl/macrosheets/_rels/sheet2.bin.rels              0 1980-01-01 00:00:00
   23 xl/drawings/_rels/drawing1.xml.rels               0 1980-01-01 00:00:00
   24 xl/sharedStrings.bin                              0 1980-01-01 00:00:00
   25 xl/worksheets/binaryIndex1.bin                    0 1980-01-01 00:00:00
   26 xl/worksheets/binaryIndex2.bin                    0 1980-01-01 00:00:00
   27 xl/worksheets/binaryIndex3.bin                    0 1980-01-01 00:00:00
   28 xl/worksheets/binaryIndex4.bin                    0 1980-01-01 00:00:00
   29 xl/macrosheets/binaryIndex1.bin                   0 1980-01-01 00:00:00
   30 xl/macrosheets/binaryIndex2.bin                   0 1980-01-01 00:00:00
   31 xl/macrosheets/binaryIndex3.bin                   0 1980-01-01 00:00:00
   32 xl/printerSettings/printerSettings1.bin           0 1980-01-01 00:00:00
   33 xl/printerSettings/printerSettings2.bin           0 1980-01-01 00:00:00
   34 xl/calcChain.bin                                  0 1980-01-01 00:00:00
   35 docProps/core.xml                                 0 1980-01-01 00:00:00
   36 docProps/app.xml                                  0 1980-01-01 00:00:00
```

As you can see, no XML files but ".bin" files but some streams disclose the presence of macros:

```
xl/macrosheets/sheet1.bin
xl/macrosheets/_rels/intlsheet1.bin.rels
xl/sharedStrings.bin
```

sharedStrings.bin is a very good indicator! Let's dump it:

```
remnux@remnux:/MalwareZoo/20220324$ zipdump.py luamccsbuaraeos.xlsb -s 24 -a | more
00000000: 9F 01 08 17 00 00 00 17  00 00 00 13 07 00 01 00  ................
```

```
00000010: 00 00 58 00 13 07 00 01  00 00 00 64 00 13 07 00  ..X........d....
00000020: 01 00 00 00 22 00 13 07  00 01 00 00 00 3A 00 13  ....".........:..
00000030: 07 00 01 00 00 00 3D 00  13 07 00 01 00 00 00 2C  ......=........,
00000040: 00 13 07 00 01 00 00 00  5C 00 13 07 00 01 00 00  ........\.......
00000050: 00 43 00 13 07 00 01 00  00 00 41 00 13 07 00 01  .C........A.....
00000060: 00 00 00 2E 00 13 07 00  01 00 00 00 4C 00 13 07  ............L...
00000070: 00 01 00 00 00 26 00 13  07 00 01 00 00 00 28 00  .....&........(.
00000080: 13 07 00 01 00 00 00 55  00 13 07 00 01 00 00 00  .......U........
00000090: 2D 00 13 07 00 01 00 00  00 29 00 13 07 00 01 00  -........).......
000000A0: 00 00 54 00 13 07 00 01  00 00 00 52 00 13 07 00  ..T........R....
000000B0: 01 00 00 00 45 00 13 07  00 01 00 00 00 4E 00 13  ....E........N..
000000C0: D9 01 00 6A 00 00 00 22  00 68 00 22 00 26 00 22  ...j...".h.".&."
000000D0: 00 74 00 74 00 70 00 22  00 26 00 22 00 73 00 3A  .t.t.p.".&.".s.:
000000E0: 00 2F 00 2F 00 6D 00 61  00 22 00 26 00 22 00 6E  ././.m.a.".&.".n
000000F0: 00 61 00 22 00 26 00 22  00 72 00 65 00 22 00 26  .a.".&.".r.e.".&
00000100: 00 22 00 73 00 74 00 61  00 22 00 26 00 22 00 75  .".s.t.a.".&.".u
00000110: 00 72 00 61 00 22 00 26  00 22 00 6E 00 74 00 22  .r.a.".&.".n.t."
00000120: 00 26 00 22 00 65 00 2E  00 63 00 22 00 26 00 22  .&.".e...c.".&."
00000130: 00 6F 00 6D 00 2F 00 44  00 6E 00 22 00 26 00 22  .o.m./.D.n.".&."
00000140: 00 69 00 35 00 22 00 26  00 22 00 4C 00 36 00 46  .i.5.".&.".L.6.F
00000150: 00 22 00 26 00 22 00 4D  00 4C 00 22 00 26 00 22  .".&.".M.L.".&."
00000160: 00 65 00 56 00 22 00 26  00 22 00 32 00 2F 00 4E  .e.V.".&.".2./.N
00000170: 00 68 00 22 00 26 00 22  00 66 00 6E 00 76 00 22  .h.".&.".f.n.v."
00000180: 00 26 00 22 00 68 00 2E  00 70 00 6E 00 22 00 26  .&.".h...p.n.".&
00000190: 00 22 00 67 00 22 00 2C  00 22 00 13 DF 01 00 6D  ."..g.".,."......m
000001A0: 00 00 00 22 00 68 00 22  00 26 00 22 00 74 00 22  ...".h.".&.".t."
000001B0: 00 26 00 22 00 74 00 70  00 22 00 26 00 22 00 73  .&.".t.p.".&.".s
```

We have URLs:

```
hxxps://manarestaurante[.]com/Dni5L6FMLeV2/Nhfnvh.png
hxxps://hondadominicana[.]com/vAXoUmZFeV2B/Nhfnvh.png
hxxps://shlokahujafilms[.]com/Xz6RyJCaHMP/Nhfnvh.png
```

Probably, the macro will loop across the three URLs and try to fetch the next stage. The payload is a DLL file. Like in Brad's diary, it's a Qakbot[2] sample!

If you would like to have a look at the macro, the easiest way is to open the XLSB file in Excel and save it as an XLSM file. You don't need to activate macros to perform this operation but do this always in a sandbox.

Now, you'll be able to address the file as usual:

```
remnux@remnux:/MalwareZoo/20220324$ zipdump.py luamccsbuaraeos.xlsm -s 4 -d | xmldump.py pretty
<?xml version="1.0" ?>
<workbook xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main" xmlns:r="http://schemas.
    <fileVersion appName="xl" lastEdited="4" lowestEdited="6" rupBuild="4505"/>
```

```
    <workbookPr/>
    <bookViews>
        <workbookView windowHeight="11160" windowWidth="20730" xWindow="-120" yWindow="-120"/>
    </bookViews>
    <sheets>
        <sheet name="Sheet" r:id="rId1" sheetId="1"/>
        <sheet name="Rgvrb" r:id="rId2" sheetId="2" state="hidden"/>
        <sheet name="Evsrg" r:id="rId3" sheetId="3" state="hidden"/>
        <sheet name="Lgle" r:id="rId4" sheetId="4" state="hidden"/>
        <sheet name="NUEVD" r:id="rId5" sheetId="5" state="hidden"/>
        <sheet name="Rc" r:id="rId6" sheetId="6" state="hidden"/>
        <sheet name="Rcc" r:id="rId7" sheetId="7" state="hidden"/>
    </sheets>
    <definedNames>
        <definedName function="1" hidden="1" name="_xlfn.ARABIC" xlm="1">#NAME?</definedName>
        <definedName name="_xlnm.Auto_Open">NUEVD!$F$1</definedName>
    </definedNames>
    <calcPr calcId="124519"/>
    <fileRecoveryPr repairLoad="1"/>
</workbook>
```

In conclusion, we have now a new file extension to keep an eye on! These XLSB files have interesting features[3]...

[1] https://isc.sans.edu/forums/diary/Qakbot+infection+with+Cobalt+Strike+and+VNC+activity/28448
[2] https://malpedia.caad.fkie.fraunhofer.de/details/win.qakbot
[3] https://analystcave.com/excel-working-with-large-excel-files-the-xlsb-format/

Xavier Mertens (@xme)
Xameco
Senior ISC Handler - Freelance Cyber Security Consultant
PGP Key
