I'm not a robot

Unprotect excel vba project without password

After step 3 you should still have open your file in the File Compression Software (7-Zip) and you should still be in the “xl” directory – then from your desktop (or the temporary folder where you saved the modified vbaProject.bin file), left click (and hold) and “drag and drop” this file into the 7-Zip xl folder, overwriting the existing file in this location. It may not be the best tool available, but based on feedback I’ve received, it has successfully helped thousands of users. But people are fallible, and VBA passwords may be lost. So give this a try, and if it doesn’t work, you can use the next method, which uses a Hex Editor to unprotect a VBA project. Another free and easy way to remove the VBA password from your Excel file is using a HEX editor. We noticed occasions when users lost vital data. The software has amazing features, which manual methods cant provide to you like the tool supports all Excel files of MS Office (xls, .xlt, .xlsb, .xlsm, .xltm, .xlam, and all). Enter 123 as the password to unlock the VB Project. Modify “DPB=” to “DPx=” (no quotes) 5. Users Queries Manual Method to Unlock VBA Project Password in Excel Workbook Modify the *.xlsm to *.zip. Open the vbaProject.bin (or vbaProject.bas) file in XVI32 6. Click New to launch the New Range dialog box In here, we can set the title, range and password for the range, then click OK. The VBA project is now protected. However, some users have reported that it didn’t work for them. Navigate to and Select the file containing the password to be removed, then click OK. The manual method, might take a bit of practice. However, this has been changed to a softer warning message. Rename the file to restore the original the .xlsx, .xlsm or .xlam file extension. The more complex the password, the longer it takes to crack. Try opening your VBA Project inside the Visual Basic Editor (use the Alt+F11 short-cut) and click OK if you receive any error messages. VBA Project binary – VBA Projects are stored as binary files within the Excel file format structure. Since the passwords are stored in different ways, the method of removing them is also different. Enter a password in the password to modify box, then click OK. Therefore, to speed things up, any software that uses this approach tends to focus on known patterns and word combinations. To apply the protection, click Review > Protect Sheet. 2. Even Microsoft recognizes that this password is not secure. Across the board, it outperforms manual alternatives. The truth is, we don’t even need an Excel password remover for the file modify password. So, before trying this, make sure you back up the file. Visual Basic for Applications (VBA) Various MS products, particularly Excel worksheet, utilize it to create functions and automate basic processes. In the Confirm Password window, re-enter the password and click OK again. Here is the link: === How it works I will try my best to explain how it works - please excuse my English. In this write-up, you will get completely free techniques to unprotect VBA password Excel workbook. Warning Message Did you notice the warning message which appeared when setting the password? Locate the vbaProject.bin file, copy it, and paste the copy outside of the zip folder (so we can make changes to this file). Just using numbers, uppercase and lowercase letters, an 8-digit password has 218,340,105,584,896 possible character combinations. Close the file with the password to be removed. Previously the message was the same as the file open password. Some of the steps are the same as the file open password, but they are repeated here for completeness. Now select the “Protection” tab and then put the new password and save the *.xlsm file. Hold the Control key and press the F key. Next, all the sheets will appear on the screen, and select the required sheets, in which you want to unlock VBA password in Excel workbook. There are many free versions available to you – we personally like using the Freeware Hex Editor XVI32. Save the file and close the HEX Editor 6. I tried this on my system with a couple of different locked VBA project Excel files, and it worked. The worksheet is now protected. So often in fact that we created the following Ultimate Guide. Select the Protection tab, tick or untick the Locked option as required, then click OK. Not only this, here we explained how to unlock Excel password VBA code without using hex editor. Just Save As a different spreadsheet). Don’t unzip the folder completely, as the zipping and unzipping process can make the Excel file unusable. If user enters the wrong password or click Cancel, this function returns 0. Here is how you can check what version of Excel you have Below are the steps to use this VB macro code to unlock VBA projects: Keep the Excel file open that has the VBA project that you want to unlock Open a new Excel file Open the VB editor in this new Excel file (You can use the keyboard shortcut ALT + F11, or you can go to the developer tab and then click on the Visual Basic icon) In the VB editor, insert a new module (this should be for this new Excel file we have opened that does not have the locked VBA project) Copy and paste the above VBA code into this newly inserted module. If that algorithm creates the same result, then it follows that it must be the same password, and Excel allows us to edit the worksheet or workbook. Malicious users can edit the file and remove the password. Run the macro code. Modify the VBAProject.bin file with the new VBAProject.bin file. Open your modified spreadsheet with Excel. So, these are two simple methods you can use to remove the password from a VBA project quickly. The highlighted section below shows where the modify and workbook protection passwords are coded: The file modify password starts with The workbook protection password starts with Next, navigate to \xl\worksheets\sheet1.xml file in the zip file. Finally, click Save on the Save As dialog box. Start by setting the lock property of a cell or range of cells. Rename your *.zip file back to *.xlsm. xlam file types. Run-time error ‘76’: file not found: The workbook is an xls file type; the macro only works on the .xlsx, .xlsm or. The download file already includes this macro and can be used to remove the VBA password. The passwords, or the encrypted versions of those passwords are stored within the VBA binary file. A user cannot open the workbook until the password has been entered. Other Excel articles you may also like: How can I programmatically remove a (known) password from an Excel VBA project? In the left-hand Project window, right-click on the VBAproject name then select VBAProject Properties… 9. Open the files with a text editor. Since I have 64-bit Excel, I have not tested the code for 32-bit Excel. I need to modify one of their excel files but it is password protected and we do not know the password. Please see the sections below for full details on how to use the tool. We can now crack passwords from each project individually. Since these passwords can easily be removed, the softer message is more accurate. Change the file extension from .XLSM to .ZIP. Place the cursor anywhere in the UnprotectVBA subroutine. In the Visual Basic Editor window, click Tools > VBA Project Properties… The VBA Project – Project Properties window opens. Why Users Should Not Choose Manual Methods? I urgently need to update this file. Whilst we have considered six different types of passwords, a standard Excel file format handles these in only three ways: XML file code – This means the file is built using mostly code written in an XML structure. This will open the Find dialog box. Workbook protection is applied in a similar way to worksheet protection, but with fewer options. The Protect Structure dialog box opens. You’ll see it’s not too scary. Many tutorials suggest using a HEX editor to remove the password. *.xlsm (Excel 2007-present macro-enabled workbook) instructions Step2 – Once you have finished step 1, simply “Save As” your file from *.xlsm format back to *.xlsb ExcelSuperSite Ultimate Guide – How to Unprotect an Excel Spreadsheet (when I have forgotten the password) Here at ExcelSuperSite, we come across many situations where our clients have locked their spreadsheets to prevent access or accidental changes being made, but then, forget or lose access (staff leave and don’t share passwords etc) to the passwords used. However, this just isn’t true. Here we will show you some users’ queries, why they want to unlock VBA project password in Excel worksheet. The file open password prevents an Excel workbook from opening until the password is entered. They claim to be able to find 80% of passwords, which is pretty good going. After that, export the VBAProject.bin file and access the file with Hex Editor. (Remember that passwords are case-sensitive.) This is good advice for file open passwords as they cannot be easily recovered. Create a new workbook and copy the code below into a standard VBA module, then run the VBAUnprotected macro. 4. Nice and simple. This is perfectly OK. Search for the term DPB (enter DPB in the Find field and click OK). Rename the file extension of your spreadsheet from *.xlsm to *.zip and click Yes when prompted “Are you sure you want to change it?”. It is now possible to see the .xlsx file extension. Step 1 – Save your macro-enabled binary workbook to the *.xlsm spreadsheet format and then follow the step-by-step instructions for that particular file type. Click Review > Protect Workbook from the ribbon. Unlock VBA password in Excel manually has no guaranteed success rate. 10. Save the file and close the hex editor. Warning Message Did you notice the warning message this time? I have tried one of these services in they past, and had good results, try them out: . With a bit of tinkering around behind the scenes, it is possible to unprotect a VBA project in Excel without the password. Tested and works on: Excel 2007 Excel 2010 Excel 2013 - 32 bit version Excel 2016 - 32 bit version Looking for 64 bit version? Non-technical users may have issues cracking VBA project passwords in Excel. Caution: If you lose or forget the password, it cannot be recovered. Extract this file to a temporary location on your computer (i.e. to your Desktop). Because of this, most user’s password-protect VBA files. The protection is only applied when the worksheet protection is active. Worksheet protection passwords prevent specific cells from being changed. The password is not yet applied to the range. I could not have coded this macro in a million years. Search for the text “DPB=” (no quotes). Modify “DPB=” to “DPx=” (no quotes) 8. The Protect Sheet window opens. In the Confirm Password box, re-enter the password, and click OK again. Lastly, you will get the confirmation message and click on the “OK” icon. In the Folder Options dialog box, click the View tab, remove the tick from hide extensions for known file types, then click OK. These transitions of ownership usually go smooth. So, let’s get start… If you’ve ever had trouble remembering the password to one of your Excel spreadsheets, you understand how aggravating it can be. There are six different passwords within Excel: File open File modify Workbook protection Worksheet protection Allow edit ranges VBA project Each of them is applied differently and serves a different purpose. Based on the above facts, Syskare VBA Password remover for Excel is clearly the best method for unprotected VBA passwords in Excel workbook. We will cover how to remove this password later in the post. Password-Find is a paid Excel password remover. As you can see, there are several stages and formulae to follow. Open the HEX editor Open the vbaProject.bin file in the HEX editor. 3. VBA projects protected with Unviewable+ (or similar software) cannot be cracked. You can try this direct VBA approach which doesn't require HEX editing. If you think that you need software to remove Excel passwords, you will be surprised by this method. You needn’t be concerned because we’ll teach you exactly what a VBA password in Excel workbook is and how to unlock an Excel VBA project password. (Remember that passwords are case-sensitive.) However, this password can easily be removed as we will see later in this post. Change the extension of the folder from .zip to .xlsm Open the Excel file and then open the VB Editor. Open your spreadsheet in XVI32 3. Please note, the macro only works with passwords created using Excel’s standard VBA password feature. Lastly, users can easily unlock VBA passwords in Excel XLSM. There are many free versions of file compression software available – we like using 7-Zip, but you can use any software of your choice. Using any File Compression software of your choice, open your “zip” file. It prevents users from changing the structure of a workbook, such as creating or renaming worksheets. We need to adjust the XML files, then insert them, back into the zip folder in the same location. 5 It is not unusual for people other than the creators/developers of Excel spreadsheets to take ownership of these tools, especially in a business environment. Now upload the VBA project Excel files by using the “Select Files” and “Select Folders” options. Due to the Single Document Interface introduced in Excel 2013, the Windows option is no longer relevant and greyed out. The workbook structure is now protected. Private Const PAGE_EXECUTE_READWRITE = &H40 Private Declare PtrSafe Sub MoveMemory Lib "kernel32" Alias "RtlMoveMemory" _ (Destination As LongPtr, Source As LongPtr, ByVal Length As LongPtr) Private Declare PtrSafe Function VirtualProtect Lib "kernel32" (lpAddress As LongPtr, _ ByVal dwSize As LongPtr, ByVal flNewProtect As LongPtr, lpflOldProtect As LongPtr) As LongPtr Private Declare PtrSafe Function GetModuleHandleA Lib "kernel32" (ByVal lpModuleName As String) As LongPtr Private Declare PtrSafe Function GetProcAddress Lib "kernel32" (ByVal hModule As LongPtr, _ ByVal lpProcName As String) As LongPtr Private Declare PtrSafe Function DialogBoxParam Lib "user32" Alias "DialogBoxParamA" (ByVal hInstance As LongPtr, _ ByVal pTemplateName As LongPtr, ByVal hWndParent As LongPtr, _ ByVal lpDialogFunc As LongPtr, ByVal dwInitParam As LongPtr) As Integer Dim HookBytes(0 To 11) As Byte Dim OriginBytes(0 To 11) As Byte Dim pFunc As LongPtr Dim Flag As Boolean Private Function GetPtr(ByVal Value As LongPtr) As LongPtr GetPtr = Value End Function Public Sub RecoverBytes() If Flag Then MoveMemory ByVal pFunc, ByVal VarPtr(OriginBytes(0)), 12 End Sub Public Function Hook() As Boolean Dim TmpBytes(0 To 11) As Byte Dim p As LongPtr, osi As Byte Dim OriginProtect As LongPtr Hook = False #If Win64 Then osi = 1 #Else osi = 0 #End If pFunc = GetProcAddress(GetModuleHandleA("user32.dll"), "DialogBoxParamA") If VirtualProtect(ByVal pFunc, 12, PAGE_EXECUTE_READWRITE, OriginProtect) 0 Then MoveMemory ByVal VarPtr(TmpBytes(0)), ByVal pFunc, osi+1 If TmpBytes(osi) &H68 Then MoveMemory ByVal VarPtr(OriginBytes(0)), ByVal pFunc, 12 p = GetPtr(AddressOf MyDialogBoxParam) If osi Then HookBytes(0) = &H48 HookBytes(osi) = &H8 osi = osi + 1 MoveMemory ByVal VarPtr(HookBytes(osi)), ByVal VarPtr(p), 4 * osi HookBytes(osi + 4 * osi) = &HFF HookBytes(osi + 4 * osi + 1) = &HE0 MoveMemory ByVal pFunc, ByVal VarPtr(HookBytes(0)), 12 Flag = True Hook = True End If End If End Function Private Function MyDialogBoxParam(ByVal hInstance As LongPtr, _ ByVal pTemplateName As LongPtr, ByVal hWndParent As LongPtr, _ ByVal lpDialogFunc As LongPtr, ByVal dwInitParam As LongPtr) As Integer If pTemplateName = 4070 Then MyDialogBoxParam = 1 Else RecoverBytes MyDialogBoxParam = DialogBoxParam(hInstance, pTemplateName, _ hWndParent, lpDialogFunc, dwInitParam) Hook End If End Function Sub VBAUnprotected() If Hook Then MsgBox "VBA Project is unprotected!", vbInformation, "*****" End If End Sub This code does not remove the original password. Save the XML files, then copy them back into the zip folders in the same locations. However, some methods outperform others in terms of handling complicated VBA passwords, the convenience of usage, and success rates. WARNING! – When messing around with password removal, it is easy to make files unreadable, so make sure you back up before doing anything. The File Modify password has now been set. Enter a password, then click OK.Note: Prior to Excel 2013, both the Structure and Window options were available. Open the zip file, then navigate to the \xl\workbooks.xml file. *.xlsm (Excel 2007-present macro-enabled workbook) 1. When you first do this you will be prompted with quite a number of Error Messages. NOTE: DO NOT close your File Compression Software (i.e. 7-Zip) after this step as you will still need to use it in the following step 9. of VBA project file extension. Copy the vbaProject.bin file in which we have made the changes, Then go back into the xl folder within the zip folder, and replace the original vbaProject.bin file with the one you copied. Open the zip folder, locate the XL folder, and double-click on it to open it. Or maybe it’s worse when it’s your file, your password and you’ve forgotten it. If a user tries to make changes to a locked cell, an error message appears: Warning Message Depending on your version of Excel, there are different warning messages when setting the password. The only way I know to open one of these files is through a brute-force attack. This document is protected from unintentional editing. Older versions: Caution: If you lose or forget the password, it cannot be recovered. You’ll see the binary information of the file being displayed in the HEX editor. Once again change the file extension from *zip to *.xlsm. Save your spreadsheet as a different workbook – “original name – unprotected.xls” (or whatever you like. As noted above, Excel files are just text files that are written in a specific format, which Excel can interpret. Sub RemoveProtection() Dim dialogBox As FileDialog Dim sourceFullName As String Dim sourceFilePath As String Dim sourceFileName As String Dim sourceFileType As String Dim newFileName As Variant Dim tempFileName As String Dim zipFilePath As Variant Dim oApp As Object Dim FSO As Object Dim xmlSheetFile As String Dim xmlFile As Integer Dim xmlFileContent As String Dim xmlStartProtectionCode As Double Dim xmlEndProtectionCode As Double Dim xmlProtectionString As String Set dialogBox = Application.FileDialog(msoFileDialogFilePicker) dialogBox.AllowMultiSelect = False dialogBox.Title = "Select file to remove protection from" If dialogBox.Show = -1 Then sourceFullName = dialogBox.SelectedItems(1) Else Exit Sub End If sourceFilePath = Left(sourceFullName, InStrRev(sourceFullName, "\")) sourceFileType = Mid(sourceFullName, InStrRev(sourceFullName, ".") + 1) sourceFileName = Mid(sourceFullName, Len(sourceFilePath) + 1) sourceFileName = Left(sourceFileName, InStrRev(sourceFileName, ".") - 1) tempFileName = "Temp" & Format(Now, " dd-mmm-yy h-mm-ss") newFileName = sourceFilePath & tempFileName & ".zip" On Error Resume Next FileCopy sourceFullName, newFileName If Err.Number 0 Then MsgBox "Unable to copy " & sourceFullName & vbNewLine _ & "Check the file is closed and try again" Exit Sub End If On Error GoTo 0 zipFilePath = sourceFilePath & tempFileName & "\" MkDir zipFilePath 'Extract the files into the newly created folder Set oApp = CreateObject("Shell.Application") oApp.Namespace(zipFilePath).CopyHere oApp.Namespace(newFileName).items xmlSheetFile = Dir(zipFilePath & "\xl\worksheets*.xml*") Do While xmlSheetFile "" xmlFile = FreeFile Open zipFilePath & "xl\worksheets\" & xmlSheetFile For Input As xmlFile xmlFileContent = Input(LOF(xmlFile), xmlFile) Close xmlFile xmlStartProtectionCode = 0 xmlStartProtectionCode = InStr(1, xmlFileContent, "") + 2 xmlProtectionString = Mid(xmlFileContent, xmlStartProtectionCode, _ xmlEndProtectionCode - xmlStartProtectionCode) xmlFileContent = Replace(xmlFileContent, xmlProtectionString, "") End If