

Windows Wallet

The audit was conducted by TECHFUND in November 2020. The code used was from master branch available on 9th November via the following repositories.

<https://bitbucket.org/ros101/quras-core/src/master/>

<https://github.com/quras-official/smartcontract-nft>

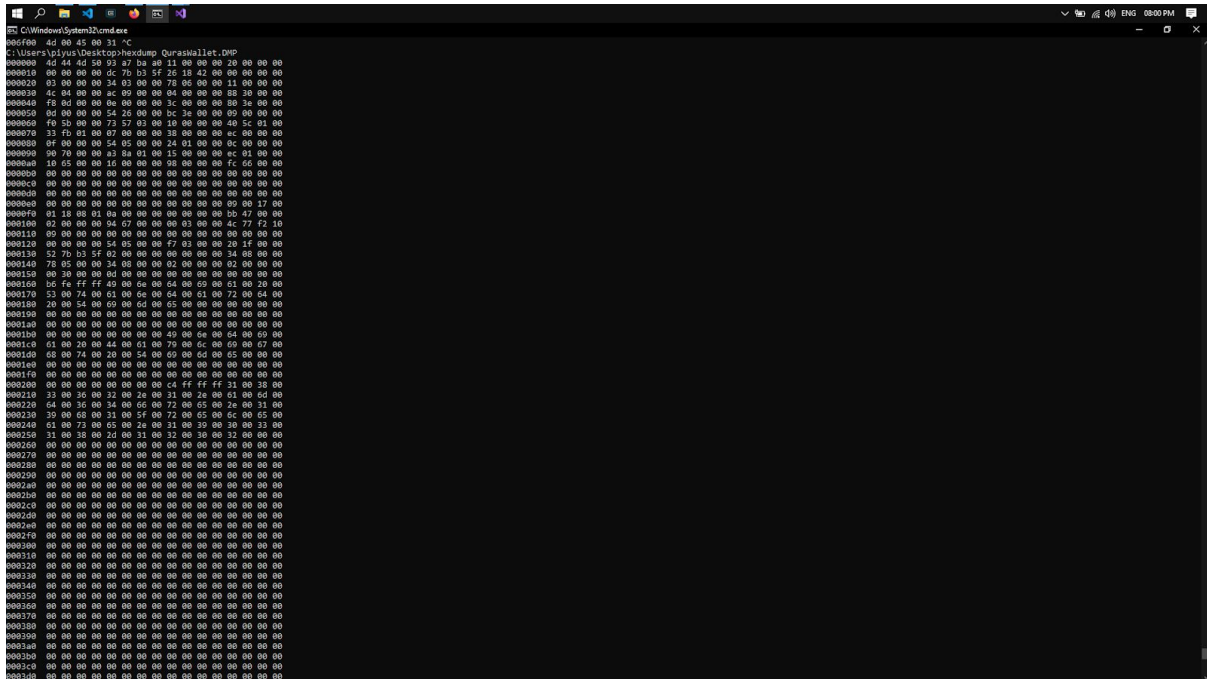
Vulnerabilities

Critical	1
High	1
Medium	3
Low	0
Note	2

We found above vulnerabilities in the code that have been described below.

1) Issue : Stealing password from running process

CRITICAL

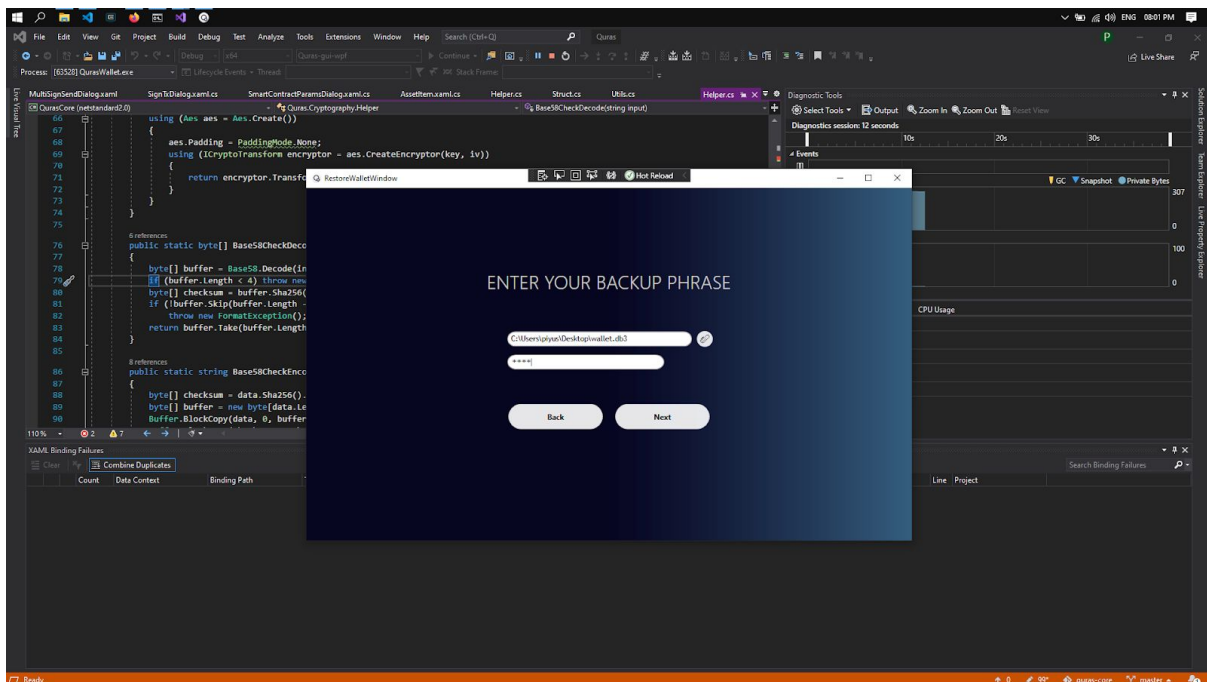


```

C:\Windows\System32\cmd.exe
00000000 4d 00 45 00 31 00 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 00 00 dc 7b b3 5f 26 18 42 00 00 00 00 00 00
00000020 03 00 00 00 24 03 00 00 78 00 00 00 11 00 00 00
00000030 4c 04 00 00 ac 09 00 00 04 00 00 00 88 3b 00 00
00000040 78 0d 00 00 0a 00 00 00 3c 00 00 00 8a 3a 00 00
00000050 0d 00 00 00 54 25 00 00 5c 3a 00 00 02 00 00 00
00000060 f0 2b 00 00 73 57 03 00 10 00 00 00 40 5c 01 00
00000070 33 7b 01 00 07 00 00 00 38 00 00 00 ec 0c 00 00
00000080 07 00 00 00 54 05 00 00 24 01 00 00 0c 00 00 00
00000090 90 70 00 00 a3 8a 01 00 15 00 00 00 ec 01 00 00
000000a0 10 05 00 00 12 00 00 00 00 00 00 00 ff 66 00 00
000000b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000e0 00 00 00 00 00 00 00 00 00 00 00 00 09 00 17 00
000000f0 01 10 00 00 01 00 00 00 00 00 00 00 00 42 00 00
00000100 02 00 00 00 54 c7 00 00 03 00 00 00 4c 77 f2 10
00000110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000120 00 00 00 00 54 05 00 00 ff 02 00 00 00 00 00 00
00000130 52 7b b3 5f 02 00 00 00 00 00 00 00 34 08 00 00
00000140 78 05 00 00 34 00 00 00 02 00 00 00 02 00 00 00
00000150 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00000160 b6 fe ff ff 40 00 00 00 6a 00 00 00 61 20 00 00
00000170 23 00 74 00 61 00 00 00 64 00 01 00 72 00 00 00
00000180 10 00 54 00 60 00 00 00 65 00 00 00 00 00 00 00
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001b0 00 00 00 00 00 00 00 00 40 00 00 00 c4 00 00 00
000001c0 01 00 20 00 44 00 01 00 79 00 00 00 69 00 00 00
000001d0 00 00 74 00 20 00 54 00 00 00 00 00 6d 05 00 00
000001e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200 00 00 00 00 00 00 00 00 c4 ff ff ff 31 00 38 00
00000210 33 00 26 00 32 00 2a 00 31 00 2a 00 61 00 6d 00
00000220 04 00 26 00 34 00 60 00 72 00 65 00 2a 00 31 00
00000230 30 00 68 00 31 00 5f 00 72 00 65 00 6c 00 65 00
00000240 01 00 78 00 65 00 2a 00 31 00 39 00 38 00 33 00
00000250 11 00 38 00 24 00 31 00 32 00 38 00 32 00 00 00
00000260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000002f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000350 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000380 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000390 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000003d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

A memory dump was taken while the process was running and user is about to login and is on the following screen



```

using (Aes aes = Aes.Create())
{
    aes.Padding = PaddingMode.None;
    using (ICryptoTransform encryptor = aes.CreateEncryptor(key, iv))
    {
        return encryptor.Transform
    }
}

public static byte[] BaseSCheckDeco
{
    byte[] buffer = BaseS8.Decode(in
    (buffer.Length < 4) throw new
    byte[] checksum = buffer.Sha256()
    if (buffer.Skip(buffer.Length -
    throw new FormatException();
    return buffer.Take(buffer.Length
}

public static string BaseSCheckEnco
{
    byte[] checksum = data.Sha256();
    byte[] buffer = new byte[data.Le
    Buffer.BlockCopy(data, 0, buffer

```

We were successfully able to extract the user password of the wallet from the memory dump. From the memory dump we extracted the strings available inside the dump file.

```
C:\Windows\System32\cmd.exe
C:\Users\piyus\Desktop>strings Quraskallet.DMP > stings_extract
```

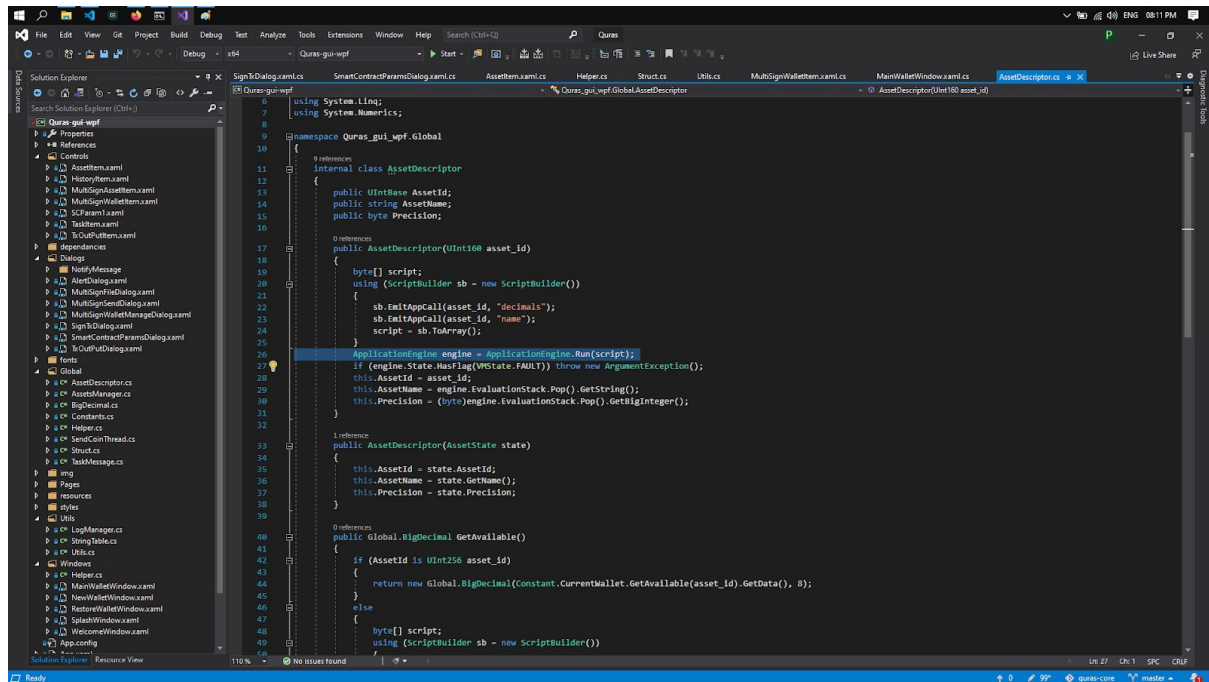
Which allowed us to view the password entered by the user directly. Any third party application on the user system can make use of similar techniques to extract the password. Using similar techniques it is also possible to grab user addresses and private keys.

```
992262 p'cZ
992263 o?U
992264 8$AU
992265 Q8?
992266 VcZ
992267 VcZ
992268 Q8?
992269 @%F[
992270 Q8?
992271 !AU
992272 !AU
992273 !AU
992274 Q8?
992275 Q8?
992276 ?cZ
992277 8\cZ
992278 HG,[
992279 8\cZ
992280 8\cZ
992281 8\cZ
992282 8\cZ
992283 ?cZ
992284 ?cZ
992285 8\cZ
992286 Password123!
992287 STR_RW_SUCCESS
992288 STR_RW_SUCCESS
992289 ?cZ
992290 STR_RW_SUCCESS
992291 STR_RW_SUCCESS
992292 Language
992293 Language
992294 Language
992295 Language
992296 Password123!
992297 Password123!
992298 0>"u
992299 8?cZ
```

2) Issue : Prevent Leakage in application engine

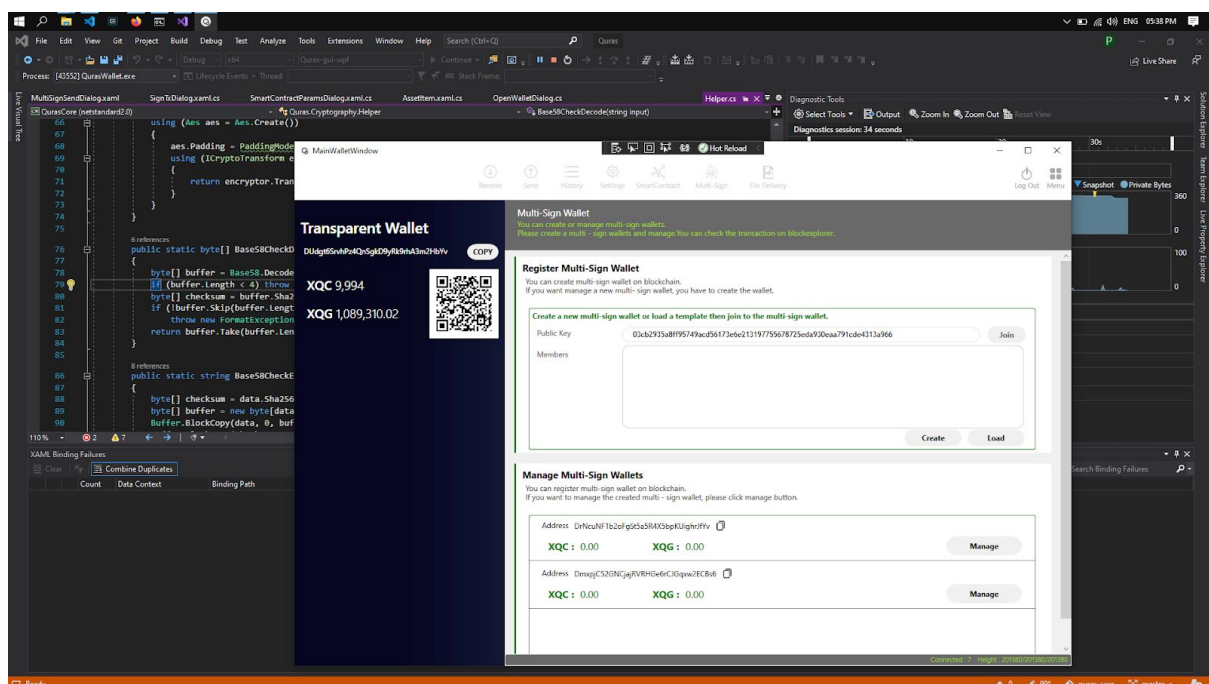
Medium

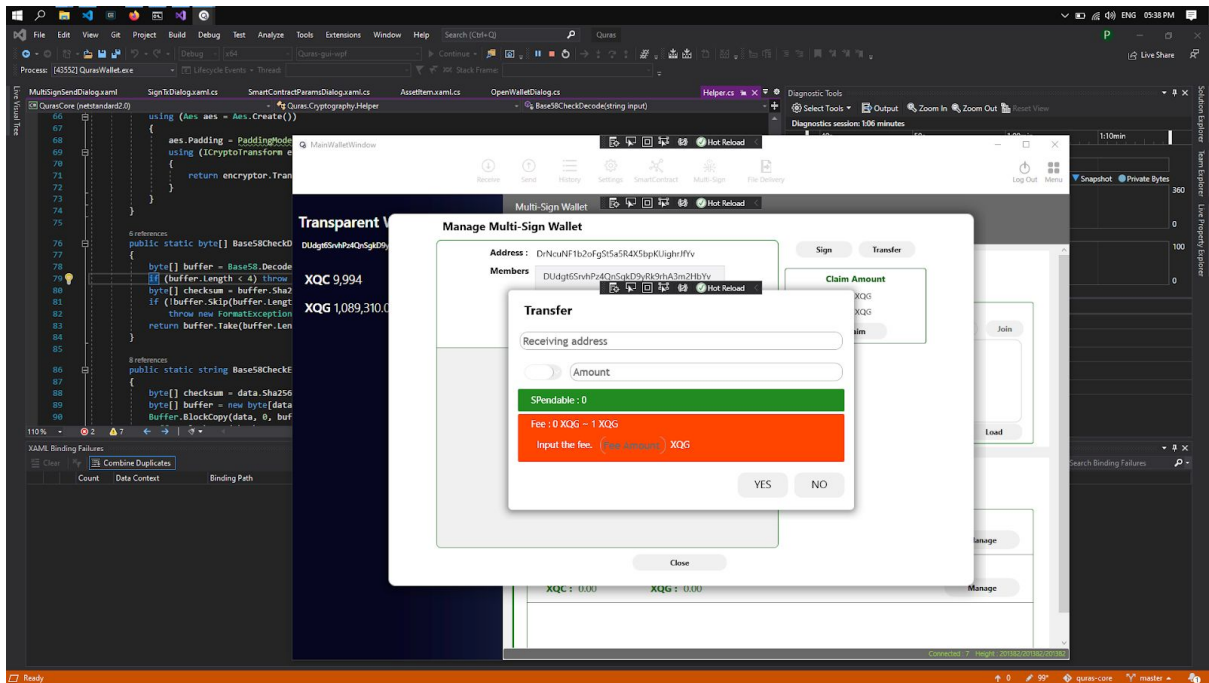
The ApplicationEngine should be wrapped by “using” statement for the “new” construct in order to prevent resource leakage.



3) Transfer byte crashes for multisign when wallet is empty

Medium





Reason :

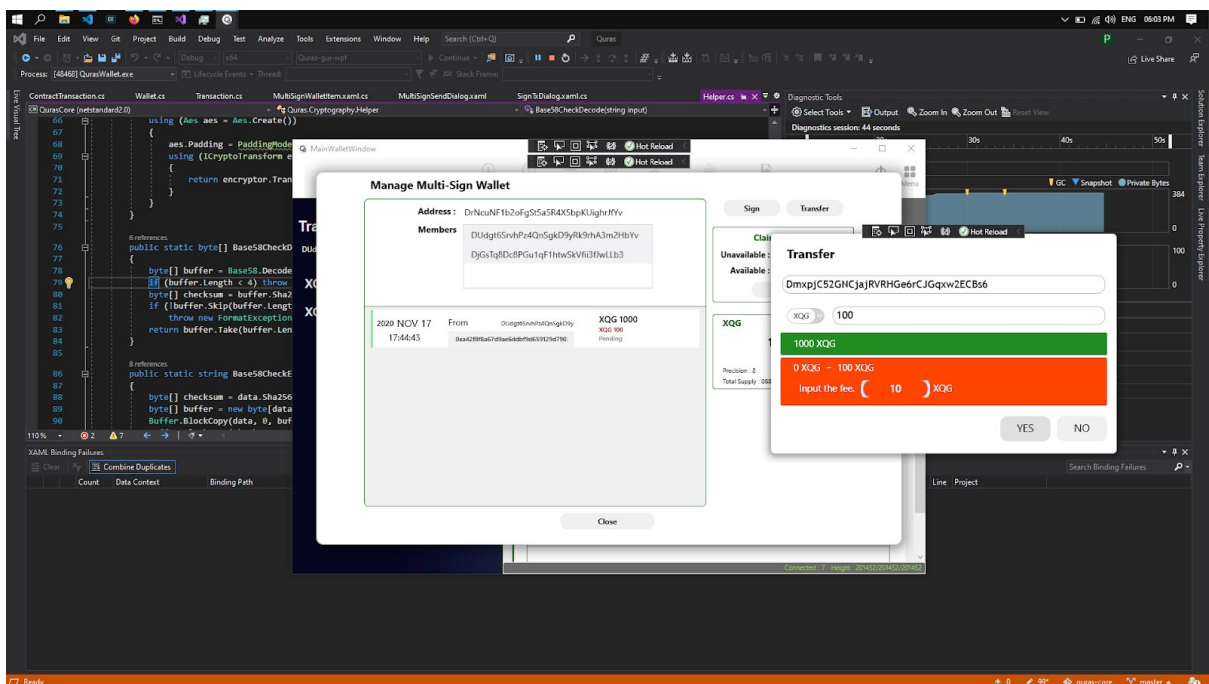
Invalid

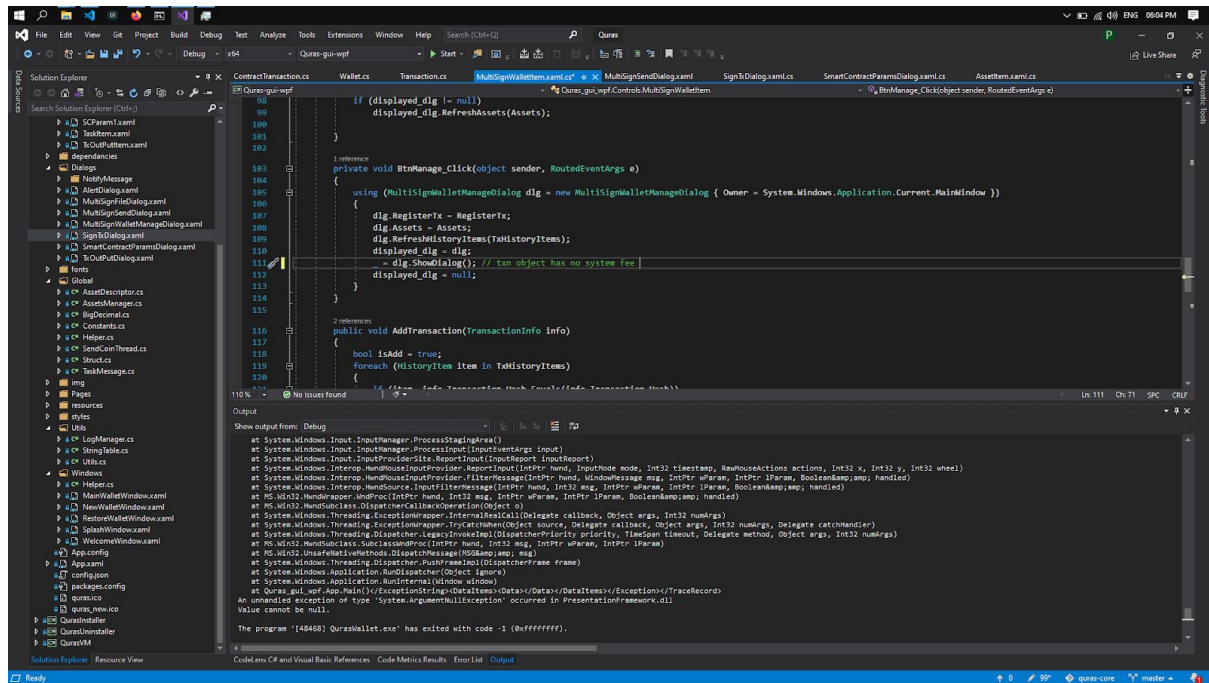
UInt256 assetId =

((AssetTypeItem)((ComboBoxItem)cmbAssetType.SelectedItem).Tag).AssetID;

4) Transfer between multisign wallets crashes the application

Medium





```

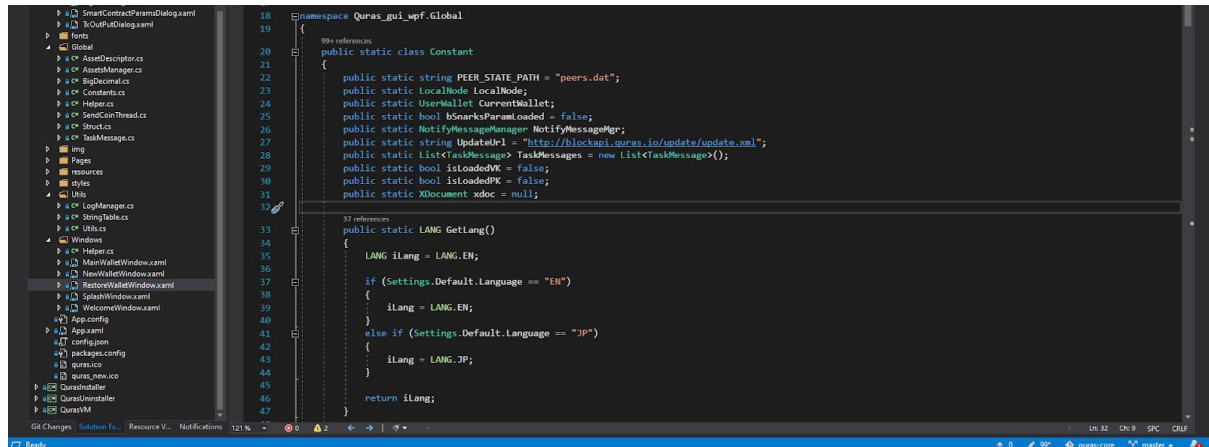
98 if (displayed_dlg != null)
99     displayed_dlg.RefreshAssets(Assets);
100
101 }
102
103 1 reference
104 private void BtnManage_Click(object sender, RoutedEventArgs e)
105 {
106     using (MultiSignWalletManagerDialog dlg = new MultiSignWalletManagerDialog { Owner = System.Windows.Application.Current.MainWindow })
107     {
108         dlg.RegisterTx = RegisterTx;
109         dlg.Assets = Assets;
110         dlg.RefreshHistoryItems(TxHistoryItems);
111         displayed_dlg = dlg;
112         = dlg.ShowDialog(); // tx object has no system fee
113         displayed_dlg = null;
114     }
115
116 2 references
117 public void AddTransaction(TransactionInfo info)
118 {
119     bool isAdd = true;
120     foreach (HistoryItem item in TxHistoryItems)
121     {
122         if (item.Transaction.Hash.Equals(info.Transaction.Hash))
123         {
124             isAdd = false;
125         }
126     }
127     if (isAdd)
128     {
129         TxHistoryItems.Add(new HistoryItem { Transaction = info.Transaction });
130     }
131 }
132
133 110% No issues found
134
135 Show output from: Debug
136 at System.Windows.Input.InputManager.ProcessStagingArea()
137 at System.Windows.Input.InputManager.ProcessInput(IInputEventArgs input)
138 at System.Windows.Interop.HwndSource.InputReport(IntPtr hwnd, InputMode mode, Int32 timestamp, RawMouseActions actions, Int32 x, Int32 y, Int32 wheel)
139 at System.Windows.Interop.HwndSource.InputReport(IntPtr hwnd, InputMode mode, Int32 timestamp, RawMouseActions actions, Int32 x, Int32 y, Int32 wheel)
140 at System.Windows.Interop.HwndSource.FilterMessage(IntPtr hwnd, IntPtr wParam, IntPtr lParam, Boolean& msg, Boolean& amp; msg; handled)
141 at System.Windows.Interop.HwndSource.FilterMessage(IntPtr hwnd, IntPtr wParam, IntPtr lParam, Boolean& msg, Boolean& amp; msg; handled)
142 at MS.Win32.HwndSource.WndProc(IntPtr hwnd, Int32 msg, IntPtr wParam, IntPtr lParam, Boolean& msg, Boolean& amp; msg; handled)
143 at MS.Win32.HwndSource.WndProc(IntPtr hwnd, Int32 msg, IntPtr wParam, IntPtr lParam, Boolean& msg, Boolean& amp; msg; handled)
144 at System.Windows.Threading.ExceptionWrapper.InternalRealCall(Delegate callback, Object args, Int32 numArgs)
145 at System.Windows.Threading.ExceptionWrapper.TryCatchWhen(Object source, Delegate callback, Object args, Int32 numArgs, Delegate catchHandler)
146 at System.Windows.Threading.Dispatcher.LegacyInvokeImpl(DispatcherPriority priority, TimeSpan timeout, Delegate method, Object args, Int32 numArgs)
147 at MS.Win32.HwndSource.WndProc(IntPtr hwnd, Int32 msg, IntPtr wParam, IntPtr lParam, Boolean& msg, Boolean& amp; msg; handled)
148 at MS.Win32.UnsafeNativeMethods.DispatchMessage(MSG& msg)
149 at System.Windows.Threading.Dispatcher.PushFrameImpl(DispatcherFrame frame)
150 at System.Windows.Application.RunDispatcher(Object ignore)
151 at System.Windows.Application.RunInternal(Window window)
152 at Quras.gui_wpf.App.Main()
153 An unhandled exception of type 'System.ArgumentNullException' occurred in PresentationFramework.dll
154 Value cannot be null.
155 The program '[48468] QurasWallet.exe' has exited with code -1 (0xffffffff).

```

5) Serve updates over https

Note

Currently the updates are being served over an http server, we highly recommend to serve updates over https. This is only as a note but is highly recommended to be followed.



```

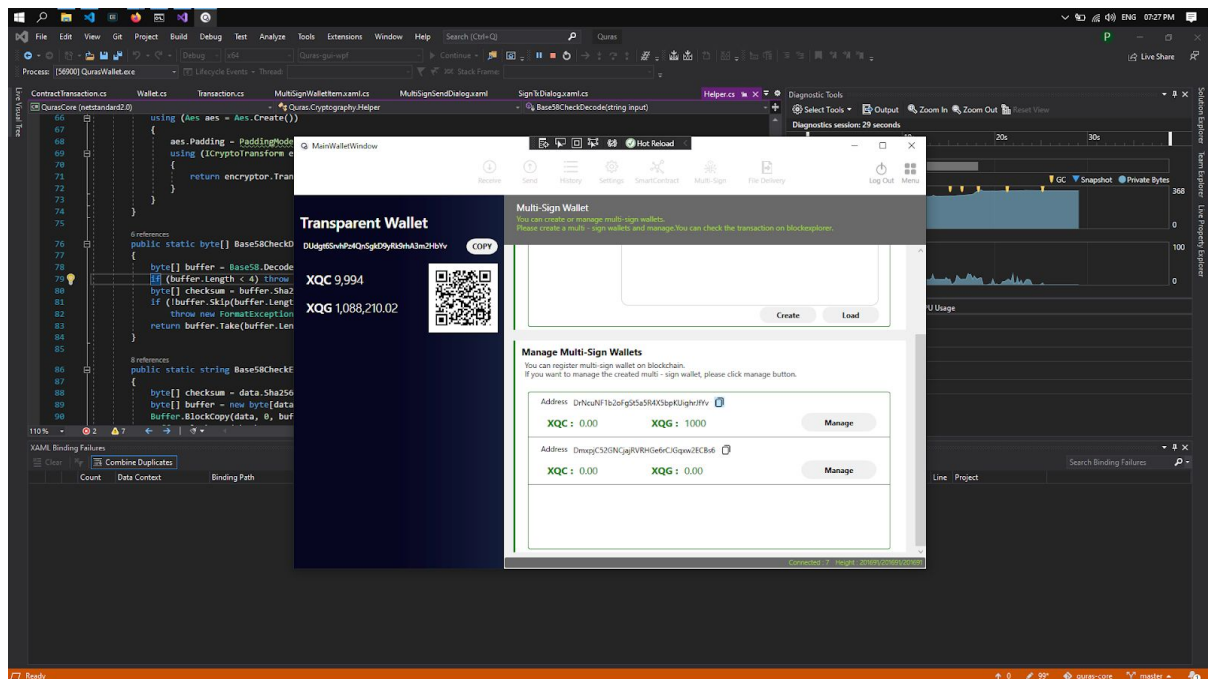
18 namespace Quras_gui_wpf.Global
19 {
20     99+ references
21     public static class Constant
22     {
23         public static string PEER_STATE_PATH = "peers.dat";
24         public static LocalNode LocalNode;
25         public static UserWallet CurrentWallet;
26         public static bool bMarksParamLoaded = false;
27         public static NotifyMessageManager NotifyMessageMgr;
28         public static string UpdateUrl = "http://blocknet.quras.io/update/update.xml";
29         public static List<TaskMessage> TaskMessages = new List<TaskMessage>();
30         public static bool bIsLoadedPK = false;
31         public static bool bIsLoadedPK = false;
32         public static XDocument xdoc = null;
33     }
34
35     37 references
36     public static LANG_GetLang()
37     {
38         LANG ilang = LANG.EN;
39
40         if (Settings.Default.Language == "EN")
41         {
42             ilang = LANG.EN;
43         }
44         else if (Settings.Default.Language == "JP")
45         {
46             ilang = LANG.JP;
47         }
48         return ilang;
49     }
50 }

```

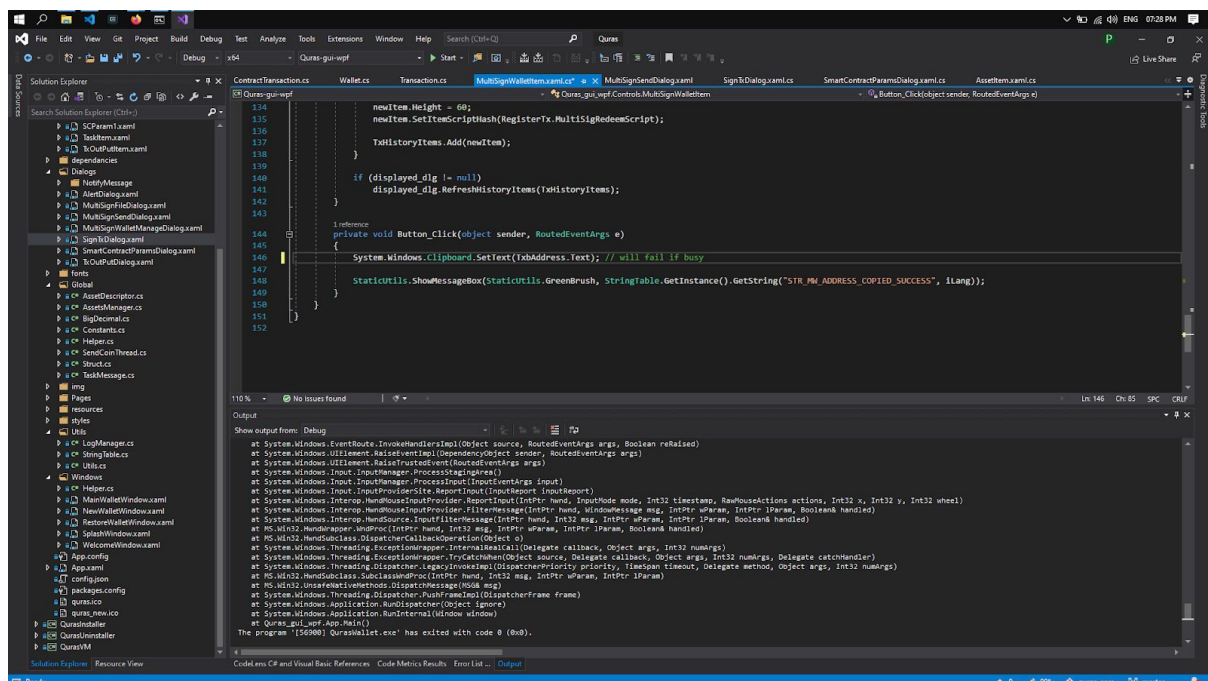
6) Denial of service

High

Application will crash if another application is accessing clipboard (*To replicate it, try clicking on the following copy button continuously and fast.)

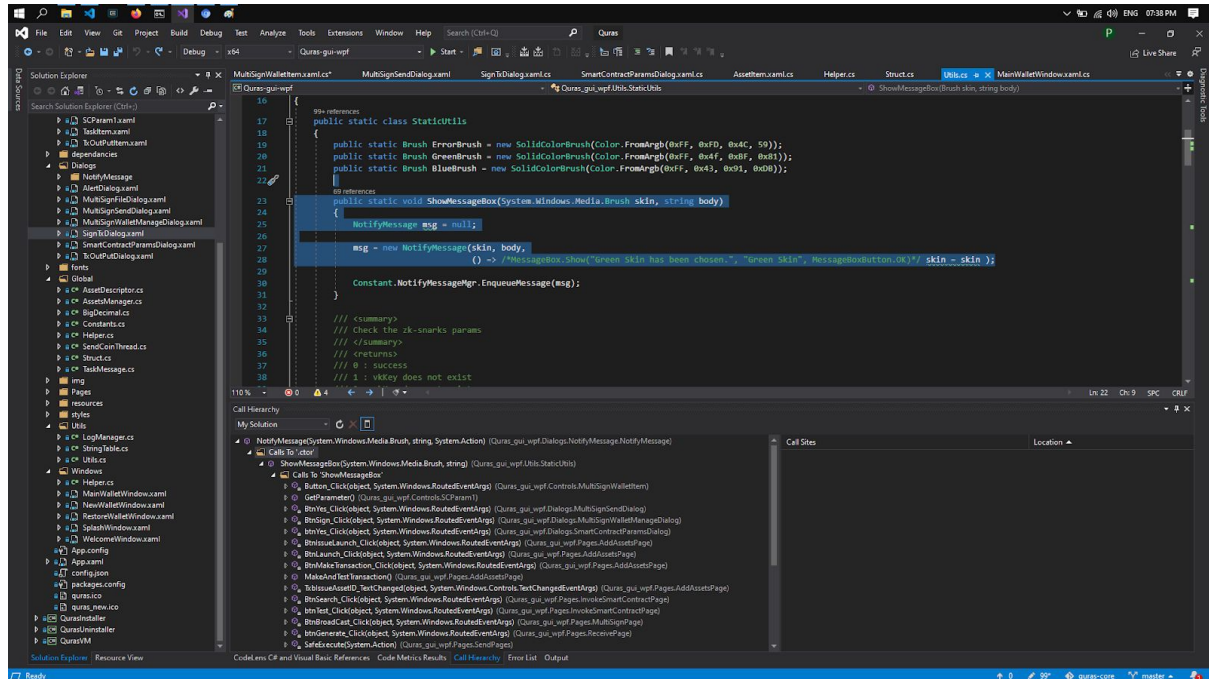


Any other application can cause denial of service to the Quras exe simply by flooding the OS clipboard when the application is in use and the user tries to copy an address.



7) Unnecessary assignment to variable

NOTE



In “notify” message, “skin” variable is assigned to self. That might not be as intended.