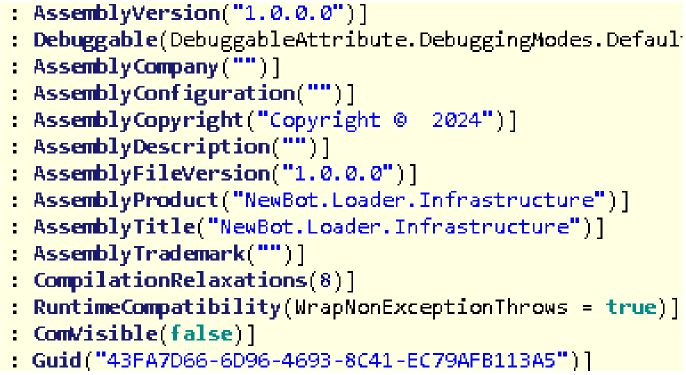
NewBot Loader. By: Jason Reaves and Joshua Platt | by Jason Reaves | Walmart Global Tech Blog | Mar, 2024

medium.com/walmartglobaltech/newbot-loader-81e2ba11c793

Jason Reaves March 12, 2024



```
: AssemblyVersion("1.0.0.0")]
: Debuggable(DebuggableAttribute.DebuggingModes.Defaul
: AssemblyCompany("")]
: AssemblyConfiguration("")]
: AssemblyCopyright("Copyright ©  2024")]
: AssemblyDescription("")]
: AssemblyFileVersion("1.0.0.0")]
: AssemblyProduct("NewBot.Loader.Infrastructure")]
: AssemblyTitle("NewBot.Loader.Infrastructure")]
: AssemblyTrademark("")]
: CompilationRelaxations(8)]
: RuntimeCompatibility(WrapNonExceptionThrows = true)]
: ComVisible(false)]
: Guid("43FA7D66-6D96-4693-8C41-EC79AFB113A5")]
```

By: Jason Reaves and Joshua Platt

Another day another new loader. During our research lately, we have discovered several new malware loaders that appear to be targeting corporate and enterprise environments.

This one calls itself NewBot Loader:

The loader is slightly obfuscated but some strings can still be seen giving a bit of insight into the capabilities.

```
<CloseShell>b__0<OpenShell>b__0<GetShell>b__0<GetInstalledEdr>b__0<GetBytes>b__0<Inject>b__0<Exec
```

The rest of the strings are loaded as single bytes:

We can recover them pretty easily though by hexlifying the entire binary and doing a regex:

```
>>> t = re.findall(r'''[a2,01]2520.{8}20..''', d)>>> tt = [(x[-2:], x) for x in t]>>> tt =
[(chr(int(x[0],16)), x[1]) for x in tt]>>> tt[0]('\x00', '12520000000002000')>>> out = "">>> for
val in tt:...   if val[1][0] == '1':...     out += '\n'...   out += val[0]...>>>
out'\n\x00\nOpening new shell...\ncmd.exe\n/k\n[\n] - \nInfo\nError\n[\n] - \nError\n[\n] -
\n{0}:{1}\nClient {0} connecting to {1}...\nUnable to perform handshake.\nClient {0} connected to
{1}...\nUnable to connect listener: \n{0} received.\nSending {0} callback...\nDisconnecting
{0}...\nDisconnected {0}...\nUnable to extract key from encoded
payload.\nSOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\nNewBot.Loader\nNewBot.Loader\nU
```

Decoded strings are appended to end of this blog, the config is mostly based on random data and a generated GUID but finding the calls to this involve going through the control flow obfuscation that is common in .NET involving overloaded class methods. We are going to briefly walk through a few relevant code blocks below:

To find where this gets used we start with the below code following the built string ‘Loader started…’:

The first call Xoshiro sets up the registry key persistence via a run key. Next a new object is created which is also where our config is setup inside the Partner function, this object is then passed to Intx which just sets the internal Fixups variable to the new object:

This gets later used and what is passed in is the C2 host and port that is also decoded from the strings:

There is also a lot of strings related to AV, EDR and analyst tools which appear to mostly come from OSINT code[1]. These strings are loaded into a string array named Hierarchy:

Later these names are retrieved in another piece of code. The manual function called just returns the previous array. Next, a few directory locations are loaded:

The loaded directories are:

```
C:\Program Files (x86)C:\Program FilesC:\ProgramData
```

The loader will then look for any sub folder containing the strings:

Decoded strings allude to encoded payload extraction. The extraction routine reads in the first 16 bytes to acquire a key that will be utilized in the decoding routine. Unfortunately, we were unable to retrieve a payload at the time of our writing.

```
   using (MemoryStream memoryStream = new MemoryStream(older, 0, older.Length))   {    byte[]
array = new byte[16];    int num = memoryStream.Read(array, 0, 16);    bool flag = num < 16;
if (flag)    {     throw new Exception(string.Concat(new string[]     {       "Unable to extract
key from encoded payload."      }));    }
```

Decoded strings

```
Opening new shell...cmd.exe/k[] -InfoError[] -Error[] -{0}:{1}Client {0} connecting to
{1}...Unable to perform handshake.Client {0} connected to {1}...Unable to connect listener:{0}
received.Sending {0} callback...Disconnecting {0}...Disconnected {0}...Unable to extract key from
encoded payload.SOFTWARE\Microsoft\Windows\CurrentVersion\RunNewBot.LoaderNewBot.LoaderUnable to
read x32 headers..text.data.pdata.reloc.rsrc.rdataAllocating
memory...NtAllocateVirtualMemoryMemory allocation failed.Allocation ended with result {0} in
{1:X}Creating thread...NtCreateThreadExCreating thread failed.Creating thread finished with
result {0} - {1}.Starting thread...NtWaitForSingleObjectUnable to start threadThread finished
with status {0}.Writing virtual memory...NtWriteVirtualMemoryUnable to write virtual
memoryWriting virtual memory finished with status {0}.ntdll.dllUnable to determine syscall for
method:Strategy cannot be null.Executing {0}...Command executed.45.15.157[.]139:1337SELECT * FROM
Win32_OperatingSystemCaptionVersionError:UnknownUnknownSELECT * FROM
Win32_ComputerSystemTotalPhysicalMemoryError:HARDWARE\DESCRIPTION\System\CentralProcessor\0PROCES
 Up TimeC:\Program Files (x86)C:\Program FilesC:\ProgramDataNo domain controller information
found.SELECT * FROM Win32_ComputerSystemDomainError:UnknownactiveconsoleADA-PreCheckahnlabanti
malwareanti-malwareantimalwareanti virusanti-virusantivirusappsenseattivo
networksattivonetworksauthtapavastavectobitdefenderblackberrycanarycarbonblackcarbon blackcheck
pointciscoampcisco
ampcounterceptcountertackcramtraycrssvccrowdstrikecsagentcsfalconcsshellcybereasoncycloramacylanc
 instinctdefendpointdefendereectrlelasticendgamef-
secureforcepointfortinetfireeyegroundlingGRRservicharfanglabinspectorivantijuniper
networkskasperskylacunalogrhythmmalwaremalwarebytesmandiantmcafeemorphisecmsascuilmsmpengnissrvom
 Alto
Networkspgeposervicepgsystemtrayprivilegeguardprocwallprotectorservicqianxinqradarqualysrapid7red

canarySanerNowsangforsecureworkssecurityhealthservicesemlaunchsvsentinelsentineloneslepliveupdatsi
 microuptycsvectrawatchguardwincollectwindowssensorwiresharkwithsecureN/AUnable to inject empty
payload.Unable to inject x86 payload.(Not supported) - Unable to inject payload to non local
process.Portable Exe injection to started...Portable Exe injection completed.Creating
section:...Section created.Starting relocationRelocating {0} to {1} with length: {2}...Resolving
imports...Loading DLLLoading injection process...Injecting payload to...Copying shellcode to
memory...Payload injection finished with wait status {0}.Unable to find handler for command:
{0}N/AN/AN/AN/AN/AUnable to get args from command: {0}Unable to decode shell action.Unable to get
model for {0}Unable to get model for {0}Unable to get model for {0}payloadTypeLoader started...
```

501a2d61bb9cdcdcbc1a77c1cf985c4d3781d60cb94380fbecac73cdbd2120baCode reuse

FastBinaryJSON

SharpEDRChecker

IOCs

```
501a2d61bb9cdcdcbc1a77c1cf985c4d3781d60cb94380fbecac73cdbd2120ba
92f3fdcbeb7175d86daaab7ac7e07db4558c0933e91552f9a50420e841a47bb3
```

```
45.15.157.]139:1337
```

Registry:

```
\Software\Microsoft\Windows\CurrentVersion\Run\NewBot.Loader
```

WMI queries:

```
SELECT * FROM Win32_OperatingSystemSELECT * FROM Win32_ComputerSystem
```

References

1:https://github.com/PwnDexter/SharpEDRChecker