Tracking StrongPity with Yara


anchorednarratives.substack.com/p/tracking-strongpity-with-yara

RJM

Alleged Turkish nation-state actor keeps infecting victims with trojanized software

Disclaimer: The views, methods, and opinions expressed at Anchored Narratives are those of the author and do not necessarily reflect the official policy or position of my employer.

Cover: Turkish delight

Introduction

Here's the latest Anchored Narrative with a follow-up story on the alleged Turkish nation-state actor StrongPity. Thanks for being a subscriber, and welcome to the new subscribers. As always, if you have any tips, comments, or great examples of disputed threat intelligence narratives for me, send them to robertjanm@anchorednarratives.com. Anchored Narratives is now also on Twitter @AnchoredNarrat1.

In the previous article about StrongPity, their history of used malware and their preferred malware infection method were covered. This article will outline how you can track a nation-state actor by leveraging the power of Yara. Yara is called the pattern matching Swiss army knife for malware researchers and is under constant development providing new features. After the last article in April 2021 on StrongPity, I developed a custom Yara signature (rule) to determine if it could track new campaigns or activity used by this nation-state actor. In this article, I will cover the creation of custom Yara rules with the well-known IDA disassembler, test them, implement them on VirusTotal Intelligence or Malpedia and perform a retrohunt. I will not cover the basics of Yara or how basic rules are constructed. Let's go!

VirusTotal Intelligence

As explained in one of the first articles on Anchored Narratives is that you can implement threat intelligence on many different sources. One of these sources is newly submitted malware samples on VirusTotal or Malpedia that are matched against custom Yara rules implemented by malware or threat analysts that track ransomware or nation-state actors. VirusTotal Intelligence (hereafter: VTI) is a paid service. The platform provides access to a large corpus of submitted malware or viruses by users or companies. The samples are stored for a longer period of time. VTI is being used by security researchers for threat hunting or finding similar files or known exploits. Researchers can basically implement *alarms* with Yara rules on known exploits, ransomware, or malware used by nation-state actors.

So after a period of StrongPity silence since the end of April 2021, last week, I received an alert on a new sample that matched my custom, Yara rule. The details of the malware sample can be seen in the figure below.

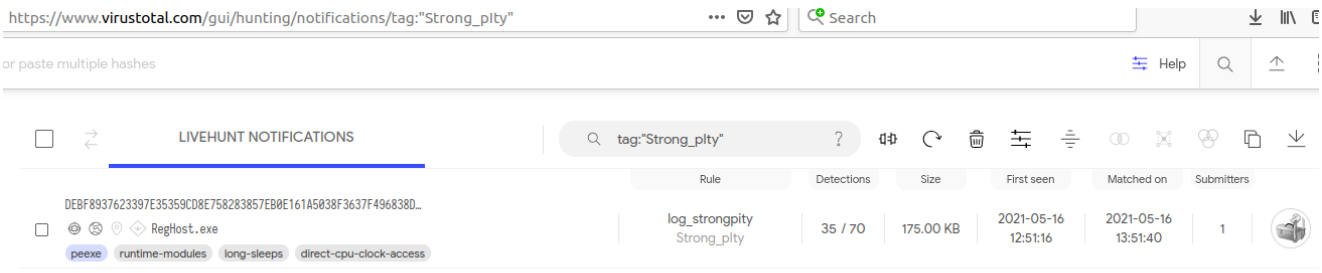

Figure 1: Screenshot VirusTotal Intelligence on StrongPity

```
SHA256:debf8937623397e35359cd8e758283857eb0e161a5038f3637f496838ddeadd0
C2: https://informationserviceslab.com/parse_ini_file.php IP:45.153.243.141
```

This time the malware sample reached out to the command and control server "`https://informationserviceslab.com`", which was not known by me yet. But what triggered the Yara rule, and how was the rule built? VirusTotal provides functionality that highlights the matched context of the Yara rule. As you might recall from the last article, the StrongPity malware has quite some interesting 'strings' that could be leveraged in a Yara rule and XOR routines to decrypt the encoded domain name, etc. Figure 2 highlights a specific hex pattern below that starts with "66 31 BC 45", reflecting such a routine. The Yara rule was triggered because of the three matching elements in the rule.

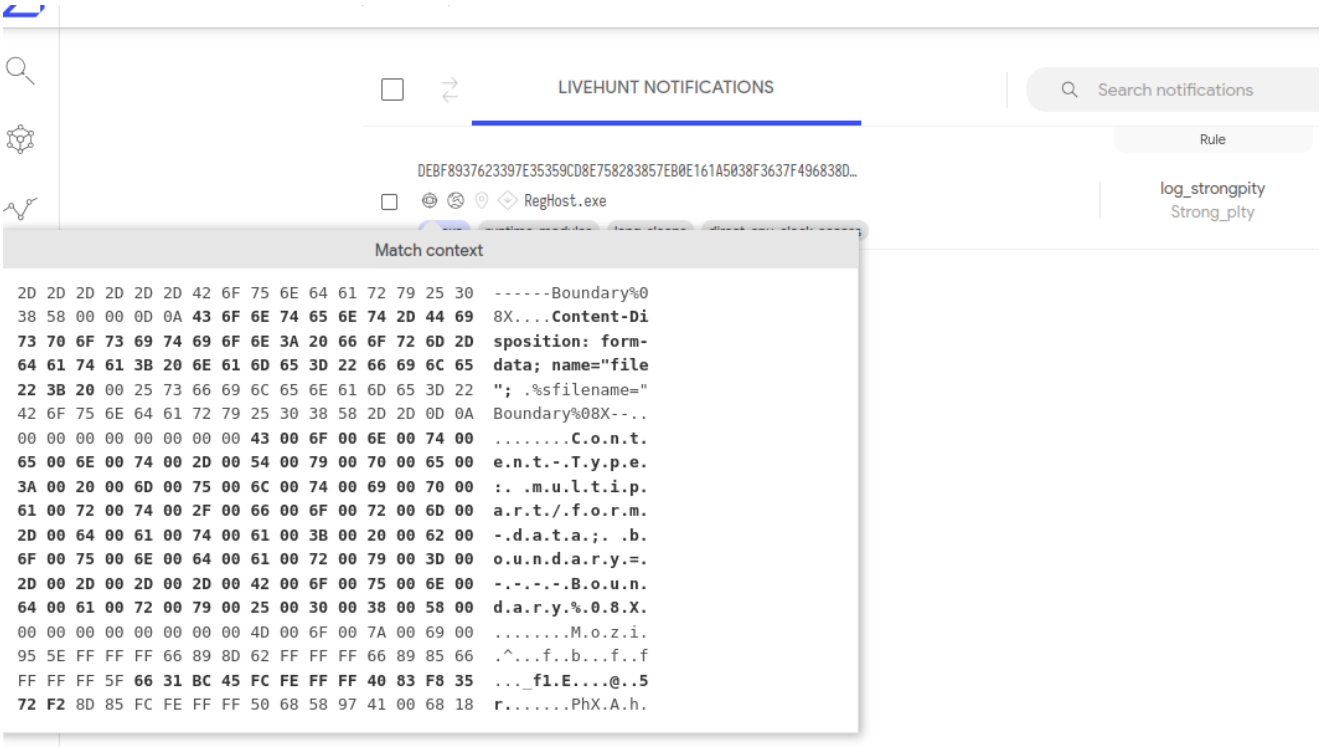

Figure 2: Screenshot of the matched context with the XOR-routine 66 31 BC 45
The Yara rule was triggered because of the three matching elements in the rule. But how was this high-fidelity Yara rule build? Let's dive into that.

Building the Yara rule

As mentioned earlier, VirusTotal Intelligence provides powerful functionality to search for similar samples. Based on earlier StrongPity reporting by ESET and CitizenLab, multiple samples were downloaded with the intent to track StrongPity.

In an in-company Yara training delivered by well-known security researcher Andreas Schuster many years ago, his advice was to focus on custom implementations of algorithms or routines by actors as a way to track adversaries. He demonstrated great examples back then to track specific Chinese nation-state actors. With that advice in mind, I started building the Yara rule to see if there was an additional way to track them. One way to assess these

discriminating artifacts is by leveraging IDA in this example. Before we start, make sure you enabled the amount of opcodes bytes (machine code) displayed in IDA7 (free). This is a method to highlight potential repetitive or certain uniqueness in relevant malware samples.

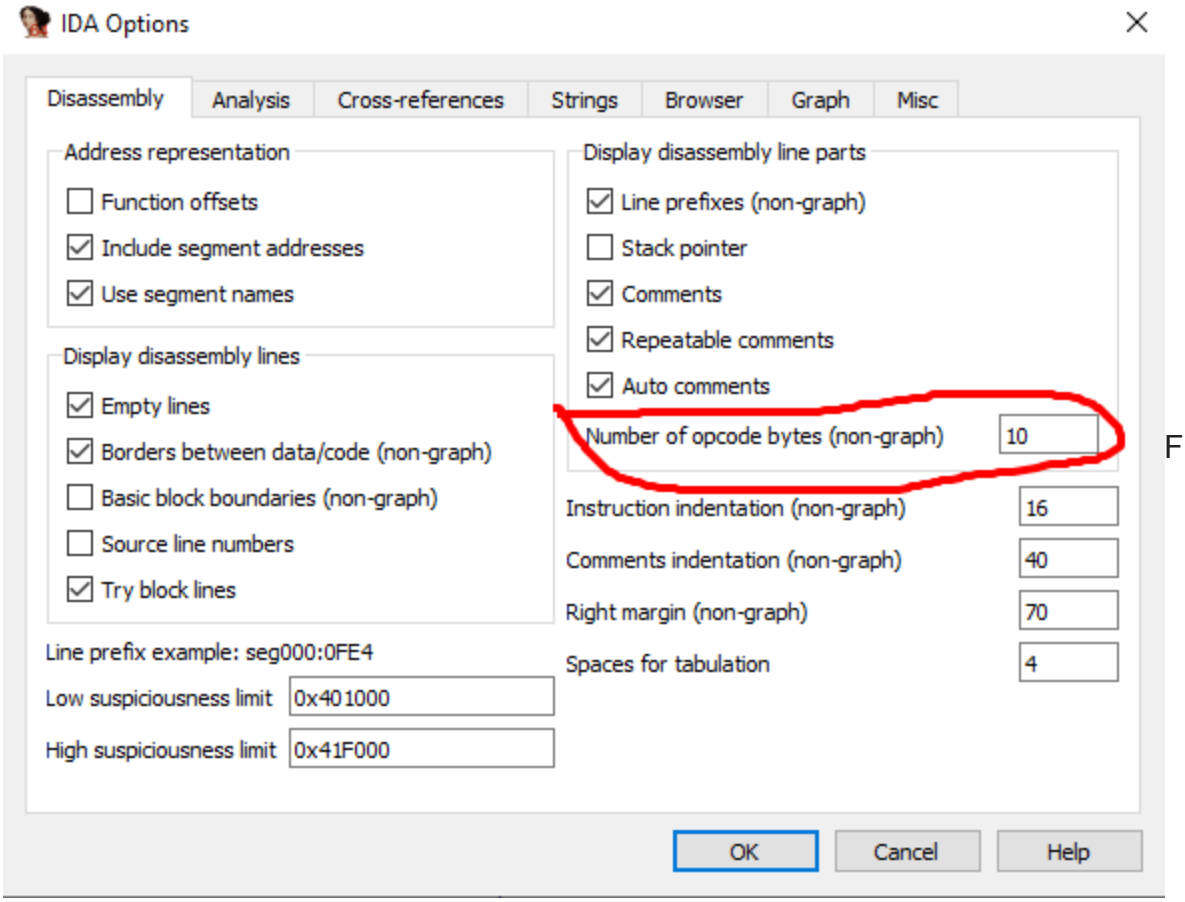

Figure 3: Screenshot of the Options tab in IDA

In this article, only several of the many StrongPity samples are covered and explained below.

`Strongpity:b9f9fb303bc605410bc1a7095da6f77d5880a1a233f849375c1aa652f9d52e1a`

StrongPity usually holds several XOR-routines with different XOR values to decode different information stored in each sample. This was covered in the earlier piece on StrongPity. So open the malware sample in IDA and determine if you could distinguish a unique routine from the actor. One of the XOR decoding routines is highlighted below.

```
.text:0040135B                          loc_40135B:                             ; CODE XREF: sub_40106A+2FE↓j
.text:0040135B 66 83 B4 45 FC FE FF FF 53       xor     [ebp+eax*2+var_104], 53h ; Logical Exclusive OR
.text:00401364 40                               inc     eax                     ; Increment by 1
.text:00401365 83 F8 35                         cmp     eax, 35h                ; Compare Two Operands
.text:00401368 72 F1                            jb      short loc_40135B ; Jump if Below (CF=1)
```

Figure 4: Screenshot XOR decoding routine

Could we make a Yara pattern of this routine? In the screenshot above, you can see an opcode byte pattern that starts with “66 83 B4 45 FC FE” and ends with the bytes “72 F1”. Let’s examine the next sample.

`StrongPity:8a7c9c4e80292bed56980d0d0fdf7c0e9693e05e3051392d5820f5037fd8f02c`

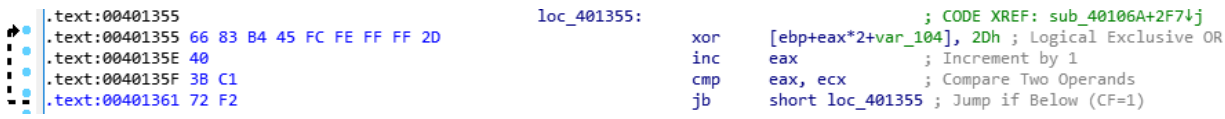

Figure 5: Screenshot another XOR decoding routine

In this different sample, you notice a similar sequence of opcode byte patterns. After this, I examined some other samples and came up with the following Yara patterns to match StrongPity samples.

```
$opcodexor1 = {66 31 ?? ?5 ?? F? FF FF 4? 83 F? ?? 72 F?}        $opcodexor2 = {66 83
B4 ?5 ?? F? FF FF [2-5] 72 F?}
```

Within Yara, you can match on strings “text” or patterns in its hexadecimal form. So with the above instructions, Yara will match against the routine found in the StrongPity malware samples. The “??” values are a wildcard and match any value. The values between the square brackets separated by a hyphen [-], are seen as a jump and match an arbitrary sequence from 4 to 5 bytes. By themselves, only those XOR patterns would trigger too many false positives. So, normally you would also leverage some telling strings found in malware samples to strengthen the Yara rule and really zoom in on a certain actor. This will reduce the number of false positives or false negatives. The StrongPity samples that I analyzed hold many of those telling strings. A very nice tool developed by Florian Roth called YarGen was leveraged to find those patterns automatically. After a bit of testing, the following StrongPity Yara rule was created, listed below, and implemented on VTI.

```
rule log_strongpity {
   meta:
      description = "Strongpity - xor routine.txt"
      author = "RJM"
      date = "2021-04-20"
   strings:
      $s1 = "Content-Disposition: form-data; name=\"file\"; " fullword ascii
      $s2 = "Content-Type: multipart/form-data; boundary=----Boundary%08X" fullword
wide
      $s3 = "SecurityHost.exe" fullword wide
      $s4 = "-CreateMutexW" fullword ascii
      $s5 =
"name=%ls&delete=WinHttpWriteDataWinHttpQueryHeadWinHttpCloseHandWinHttpReceiveReWinHt
 ascii
      $s6 = "Windows Security Host" fullword wide
      $opcodexor1 = {66 31 ?? ?5 ?? F? FF FF 4? 83 F? ?? 72 F?}
      $opcodexor2 = {66 83 B4 ?5 ?? F? FF FF [2-5] 72 F?}

   condition:
      ( uint16(0) == 0x5a4d and filesize < 200KB and ( 3 of them )
      ) or ( all of them )
}
```

Testing the StrongPity Yara rule

As I wrote earlier, I downloaded 29 StrongPity samples previously via the similarity search functionality in VirusTotal Intelligence. So let's test if the rule is reliable enough to detect the samples. All the samples are stored in different directories underneath the "*apt/Turkey/Strongpity*" folder.

```
:~/samples/yarGen/yara_rules$ yara -r  strongpity_2.yar -s ../../apt/Turkey/Strongpity |grep $\opcode
0x764:$opcodexor1: 66 31 BC 45 FC FE FF FF 40 83 F8 35 72 F2
0x75a:$opcodexor2: 66 83 B4 45 FC FE FF FF 5B 40 3B C1 72 F2
0x8b0:$opcodexor2: 66 83 B4 75 68 FF FF FF 24 46 83 FE 2E 72 F1
0x75b:$opcodexor2: 66 83 B4 45 FC FE FF FF 53 40 83 F8 35 72 F1
0x8b0:$opcodexor2: 66 83 B4 75 68 FF FF FF 26 46 83 FE 2E 72 F1
0x87d:$opcodexor2: 66 83 B4 75 74 FF FF FF 51 46 3B F0 72 F2
0x8b3:$opcodexor1: 66 31 84 75 68 FF FF FF 46 83 FE 2E 72 F2
0x8b3:$opcodexor1: 66 31 84 75 68 FF FF FF 46 83 FE 2E 72 F2
0x90b:$opcodexor1: 66 31 84 75 64 FF FF FF 46 83 FE 31 72 F2
0x913:$opcodexor2: 66 83 B4 75 64 FF FF FF 56 46 3B F0 72 F2
0x755:$opcodexor2: 66 83 B4 45 FC FE FF FF 2D 40 3B C1 72 F2
0x8b7:$opcodexor2: 66 83 B4 75 68 FF FF FF 5A 46 3B F0 72 F2
0x786:$opcodexor1: 66 31 8C 45 EC FE FF FF 40 83 F8 32 72 F2
```

Figure 5: Screenshot of matching opcodes in StrongPity samples

Figure 5 highlights Yara matches on the decoding routines found in the IDA disassembler. Now let's see how many samples are detected with the manually created Yara rule.

```
:~/samples/yarGen/yara_rules$ yara -r  strongpity_2.yar -s ../../apt/Turkey/Strongpity |grep Strongpity
log_strongpity ../../apt/Turkey/Strongpity/vt_hit/debf8937623397e35359cd8e758283857eb0e161a5038f3637f496838ddeadd0
log_strongpity ../../apt/Turkey/Strongpity/Similar/d187ef8dea351f0f6aec3e13eff52e29fad574d147508d4dd3ba4da71eb9d63a
log_strongpity ../../apt/Turkey/Strongpity/Similar/1887977dc8ea476b5ddacccfe74e6c630222bfff1c7888eef08ce0e0c4d0d12f
log_strongpity ../../apt/Turkey/Strongpity/Similar/b9f9fb303bc605410bc1a7095da6f77d5880a1a233f849375c1aa652f9d52e1a
log_strongpity ../../apt/Turkey/Strongpity/Similar/233358861a4f8f3f100baa446655c625212503126ac23d0bb67022bd6e5b5d7d
log_strongpity ../../apt/Turkey/Strongpity/Similar/786c58acaf7a1354b0038f34adec8a46235059b8f3e87a47197f008446a5c757
log_strongpity ../../apt/Turkey/Strongpity/strongpity_twit
log_strongpity ../../apt/Turkey/Strongpity/Similar/2b26f4ce23dea823f4f7f8daf4c81550855068a4042bc150dfb71344f74b6f79
log_strongpity ../../apt/Turkey/Strongpity/Similar/8a7c9c4e80292bed56980d0d0fdf7c0e9693e05e3051392d5820f5037fd8f02c
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/bd25fd23dba6fd93443bac22087f784629fec614a86a726f578f0a02bbdbe3e3
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/a1ce1b78cc1a9d6092b086f2d0796cde519033ec0935d9cecdea86b6cda87882
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/2929fec51f78330630495673bc7694ec4c244f80fd50ede70fb33cfbf9e73e23
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/e843af007ac3f58e26d5427e537cdbddf33d118c79dfed831eee1ffcce474569
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/631cebb059f7eb8fb11afb92abdcb12587f2099c8fed14dd4046db200593a8d3
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/c72bf8537fc189b81855666d7f59ad8e24011c735921a15932275757a485e7a4
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/fbd66a4f385e8c573c51c19a49c7e9c2ffa1639f4648721591b7ea0af845a313
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/12e670dc36ac50e86a58f759fa4a5de25e574227a19e1942aaa788c82540a910
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/1af0958f8590b626bedfcd1972cd3ea49d9576db86f1e768e5520f9615d01a19
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/8b33c78f8f346b6562a6314a8a223d160b79407d6ccdd004a4fa915018df2fc3
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/9e2f961d212747daae69c6bc9062ed88898ea7ae05cac440244162b27a706231
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/1939d0aa2b5560ea144365e60c911e15fe37ab528b74bda43ad8ead9935b6896
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/8844d234d9e18e29f01ff8f64db70274c02953276a2cd1a1a05d07e7e1feb55c
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/5028a2040440071741 62afcac28a4862fd70cda897d6f15864442bb2c3b8e105
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/fea02a9a6c850c8ab0285affb2ad157511614b4aa78e84802ddac2c3f0279205
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/e867bcac8bf13982ba5781f0d7863cdae50704a49dfd582557752d838b8a0b4b
log_strongpity ../../apt/Turkey/Strongpity/Similar_small/4ee465d58613c03c15c0e92728bba76a065149d4773a1ce59c76d414d70fb190
log_strongpity ../../apt/Turkey/Strongpity/Similar_downloads/dfd0f4b821438d8a9277728e42ab58bdc2667aa7173892ffd6ede75a5d5645f5
log_strongpity ../../apt/Turkey/Strongpity/Similar2/057e27d215f4930469417bfd5fec41b193c85ac9275a1ae5594fcbab68c23ed7
log_strongpity ../../apt/Turkey/Strongpity/Similar2/dfd0f4b821438d8a9277728e42ab58bdc2667aa7173892ffd6ede75a5d5645f5
                        :~/samples/yarGen/yara_rules$ yara -r  strongpity_2.yar -s ../../apt/Turkey/Strongpity |grep Strongpity |wc -l
29
```

Figure 6: Screenshot all the malware samples

The StrongPity Yara rule matched exactly 29 samples. After testing it on other malware samples, the rule seems solid and did not produce any false positives. So basically, the next step is to implement the Yara rule as a live hunt rule in VirusTotal Intelligence. This resulted in the match of a brand new StrongPity sample covered in the introduction of this article. The reason for implementing it first as a live hunt rule is to assess if it generates false positives or false negatives on that platform. I have made many mistakes in the past where I deployed a too generic rule. So testing and validating are very important.

Retrohunting StrongPity

For paid subscribers or researchers, VirusTotal Intelligence provides a capability to search across multiple samples. For regular users, the corpus of malware samples that are searched is 3 months. If you are a pro user, you can search the entire corpus for the past 12

months. Retrohunts are usually limited and expensive searches. Similar searches can be conducted on Malpedia or other platforms supporting Yara rules and storing historical malware samples.

So after I verified the rule, the StrongPity retrohunt was executed. After a while, the rule matched 127 StrongPity samples in the past 12 months. Only 9 in the past 3 months.

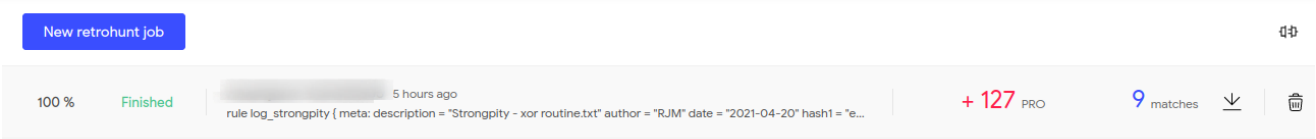

Figure 7: Screenshot of the StongPity Retrohunt and hits

Upon inspection, none of the 9 samples that matched were false positives. Maybe someone with VTI pro access can run the StrongPity retrohunt to obtain the full results for further assessment? Reach out if you were able to obtain all samples and if there were any false positives.

Conclusion

Yara is a good way to distinguish between malware of a given family and other files based on known patterns and obtain intelligence on relevant nation-state actors or ransomware actors. In this article, I outlined a method to build a discriminating Yara rule that was leveraged to track a known nation-state adversary, in this case, the alleged Turkish actor group StrongPity. The rule was also used to perform a successful retrohunt. With the StrongPity Yara rule, we can track new malware samples of the actor and collect many historical samples for the past year(s). There could also be other routines leveraged by StrongPity that discriminate and provide even better insights into how big their spyware campaigns are. That said, finding 127 malware samples submitted to VirusTotal in the past 12 months with a custom build Yara rule indicates the rule has high fidelity and that the actors behind StrongPity must be collecting massive amounts of personal data. I'm curious about how this data collected from different victims in countries around the globe is processed.

The StrongPity actor remains very active and obtains new infrastructure per campaign, it seems. The next threat actor that will be covered as an Anchored Narrative is from the Asia-Pacific region. Until next time and reach out if you want to share different or additional insights or feedback.