VIRUSTOTAL

Ransomware in a global context

MONDAY, OCTOBER 04, 2021 VICENTE DÍAZ LEAVE A COMMENT

Today we are proud to announce our very first VirusTotal Ransomware Activity Report. This initiative is designed to help researchers, security practitioners and the general public better understand the nature of ransomware attacks by sharing VirusTotal's visibility.
We are also organizing a series of webinars describing the main findings of our research, so please join us for the session that works better for you:

VIRUSTOTAL
RANSOMWARE IN A GLOBAL CONTEXT
LAUNCH WEBINARS

OCT 5, 2021
7:00 UTC
15:00 SGT
18:00 AEDT

OCT 6, 2021
16:00 UTC
9:00 PDT

OCT 7, 2021
In a Public Sector Context
9:00 UTC
11:00 CEST

October 5th (APAC-friendly timezone): https://bit.ly/3lZuS3K
October 6th (Americas and EMEA-friendly timezone): https://bit.ly/3APK49S

October 7th (Public Sector edition): https://bit.ly/3ERXioS

We encourage you to read the full report, but below you can find some of the **main findings**:

- Since 2020, users from more than 140 countries have submitted ransomware samples to VirusTotal.
- During this time, at least 130 different ransomware families have been active.
- Israel, South Korea, Vietnam, China, Singapore, India, Kazakhstan, Philippines, Iran and the UK are the 10 most affected territories based on the number of submissions to VirusTotal.
- Activity among the most spread ransomware families comes and goes, but there is a baseline of activity of around 100 not-so-popular ransomware families that never stops.
- According to our observations, it seems that in most cases attackers prepare fresh new samples for their campaigns.
- In July 2021 we observed a wave of the new variant of Babuk ransomware.
- GandCrab was the most active family in early 2020, before its prevalence decreased dramatically in the second half of the year.

You can download the full report here.

Now, how to transform all this information into something actionable we could use to protect from ransomware attacks? In this blog post we will not go over the content of the report itself. We want to discuss ideas we can use to proactively defend ourselves.

Monitoring Ryuk campaigns

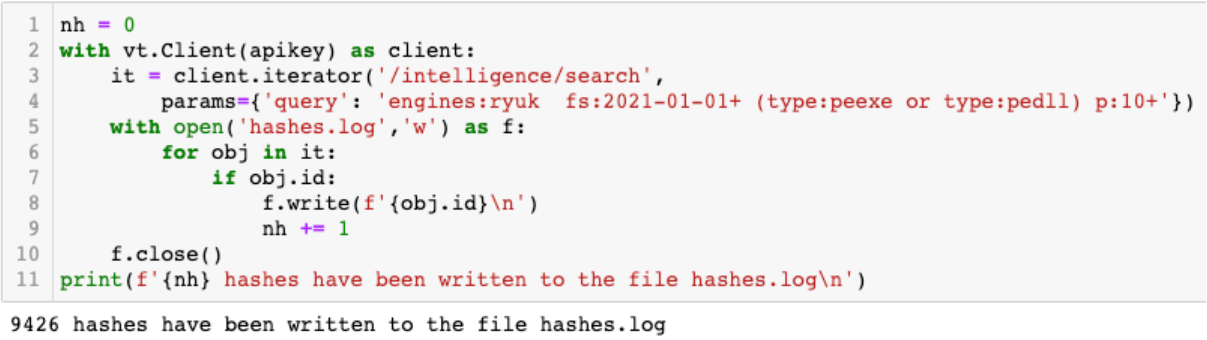
The report contains insights on ransomware families and artifacts associated with their attacks. As an example, we can use this information to prioritize enforcing new security policies in our network based on the most active families.

For instance, a first approach would be checking if any sample related to these campaigns has landed in our network. Let’s use the Ryuk ransomware family as an example. The following VirusTotal Intelligence query will help us find Ryuk PE samples with at least 10 AV detections submitted since January 2021:

"engines:ryuk fs:2021-01-01+ (type:peexe or type:pedll) p:10+"

Given this query returns more than 9k results, we can use the VT API or the VT-PY programming interfaces. An easy way to do it would be using Jupyter Notebooks to create our custom report using some fancy graphics. We have created a couple of notebooks here and here implementing some examples using the VT-PY interface we will describe below.

Let’s use one of the notebooks as an example where we want to list all the hashes submitted during a specific period of time related to the ransomware family we are monitoring. We basically iterate the results of the VT Intelligence query, resulting in 9426 hashes we will store in a log file.



```
nh = 0
with vt.Client(apikey) as client:
    it = client.iterator('/intelligence/search',
        params={'query': 'engines:ryuk  fs:2021-01-01+ (type:peexe or type:pedll) p:10+'})
    with open('hashes.log','w') as f:
        for obj in it:
            if obj.id:
                f.write(f'{obj.id}\n')
                nh += 1
    f.close()
print(f'{nh} hashes have been written to the file hashes.log\n')
```

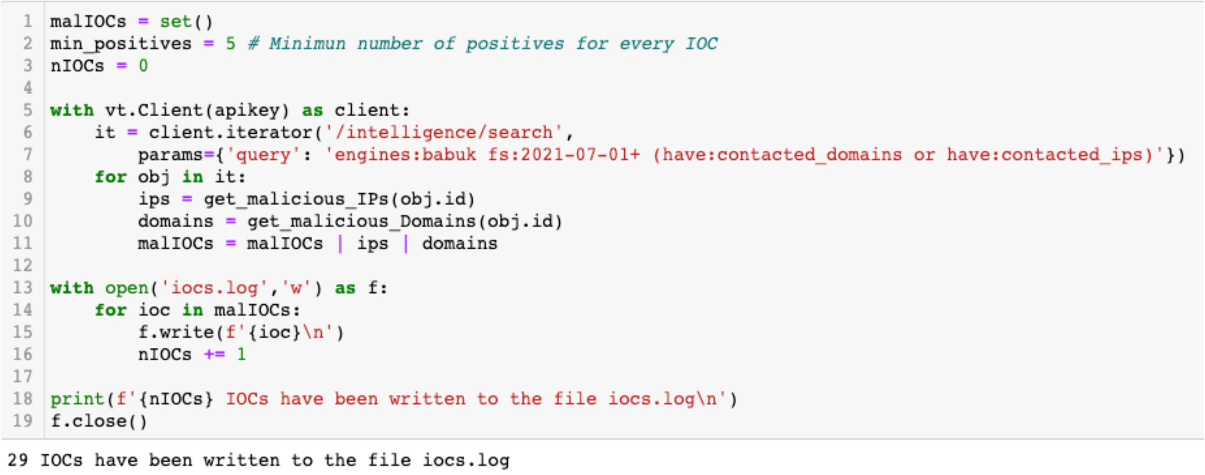
```
9426 hashes have been written to the file hashes.log
```

Monitoring Babuk

Another idea would be to collect IOCs (Indicators Of Compromise) related to these campaigns, in this case identified as malicious by at least 5 antivirus engines. Here we could get all the suspicious URLs, domains and IP addresses contacted by the malware samples, or we could retrieve URLs used at different stages of the attack. This can be done with the following VT Intelligence query:

"engines:babuk fs:2021-07-01+ (have:contacted_domains or have:contacted_ips) p:5+"

For instance, the second Jupyter notebook searches for all the domains and IP addresses contacted by Babuk since July 2021 with at least 5 positives. We can later use these IOCs to block their access in our EDR, firewall or web proxies, avoiding any attempt to contact them.



```
malIOCs = set()
min_positives = 5 # Minimun number of positives for every IOC
nIOCs = 0

with vt.Client(apikey) as client:
    it = client.iterator('/intelligence/search',
        params={'query': 'engines:babuk fs:2021-07-01+ (have:contacted_domains or have:contacted_ips)'})
    for obj in it:
        ips = get_malicious_IPs(obj.id)
        domains = get_malicious_Domains(obj.id)
        malIOCs = malIOCs | ips | domains

with open('iocs.log','w') as f:
    for ioc in malIOCs:
        f.write(f'{ioc}\n')
        nIOCs += 1

print(f'{nIOCs} IOCs have been written to the file iocs.log\n')
f.close()
```

```
29 IOCs have been written to the file iocs.log
```

Distribution vector and spreading

It is always a good idea protecting ourselves at the initial stages of an attack. We can monitor the infrastructure used for distribution of any campaign making use of our itw ("in the wild") tag. Additionally, we can also search for files executing or containing malware related to the campaign we monitor. These queries will help us to block any malicious infrastructure as well as to detect samples distributing the malware we monitor. This can be done with the following VT Intelligence query:

"engines:gandcrab fs:2020-02-01+ fs:2020-05-01- (type:peexe or type:pedll) have:in_the_wild"

VTI Search Link

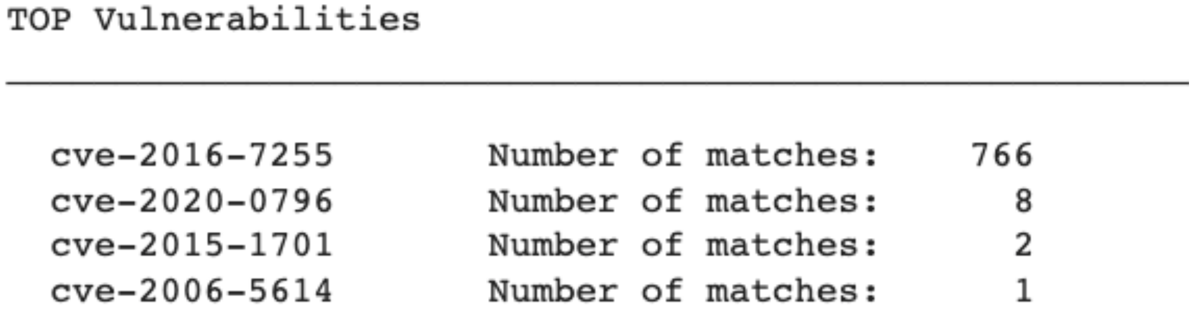
We have also created a script available in one of the aforementioned Jupyter Notebooks showing the list of distribution vectors related to Gandcrab ransomware.

Another interesting angle is understanding what exploits a specific threat campaign is using for spreading. We can do that using the tag:exploit modifier in our VT Intelligence query. For example:

"engines:gandcrab fs:2020-02-01+ fs:2020-05-01- (type:peexe or type:pedll) tag:exploit"

This modifier would return those samples that are suspected to contain an exploit. This can be used to list the top countries that submitted samples related to this particular malware family containing exploits.

The same approach can be taken on a typical vulnerability management use case. One of the Jupyter notebooks provides the top list of exploited vulnerabilities related to a malware family.



```
TOP Vulnerabilities
____________________________________________________________

  cve-2016-7255       Number of matches:    766
  cve-2020-0796       Number of matches:      8
  cve-2015-1701       Number of matches:      2
  cve-2006-5614       Number of matches:      1
```

Are we in trouble?

Another common approach is checking if our brand has been abused in any phishing campaign or if our infrastructure hosted any component of the attack. The following VT Intelligence query will search from any embedded domain or URLs used in recent Cerber campaigns, including URLs used for storing malware samples (itw urls):

"engines:cerber fs:2021-06-01+ (embedded_domains:my_domain OR embedded_urls:my_domain OR itw:my_domain)"

What’s next?

The information provided by the VirusTotal community can be used to proactively monitor and protect against ransomware attacks. Some additional ideas on how to use VirusTotal in this direction can be found below:

- Global Threat Intelligence. Once we know what are the most common ransomware signatures and its generic behavior, we can use this information to monitor future samples, for instance:

"p:10- fs:2021-09-01+ (engines:ransom or engines:crypto) AND tag:persistence and tag:detect-debug-environment AND tag:checks-network-adapters AND tag:long-sleeps AND tag:direct-cpu-clock-access"

VTI Search Link

This query:

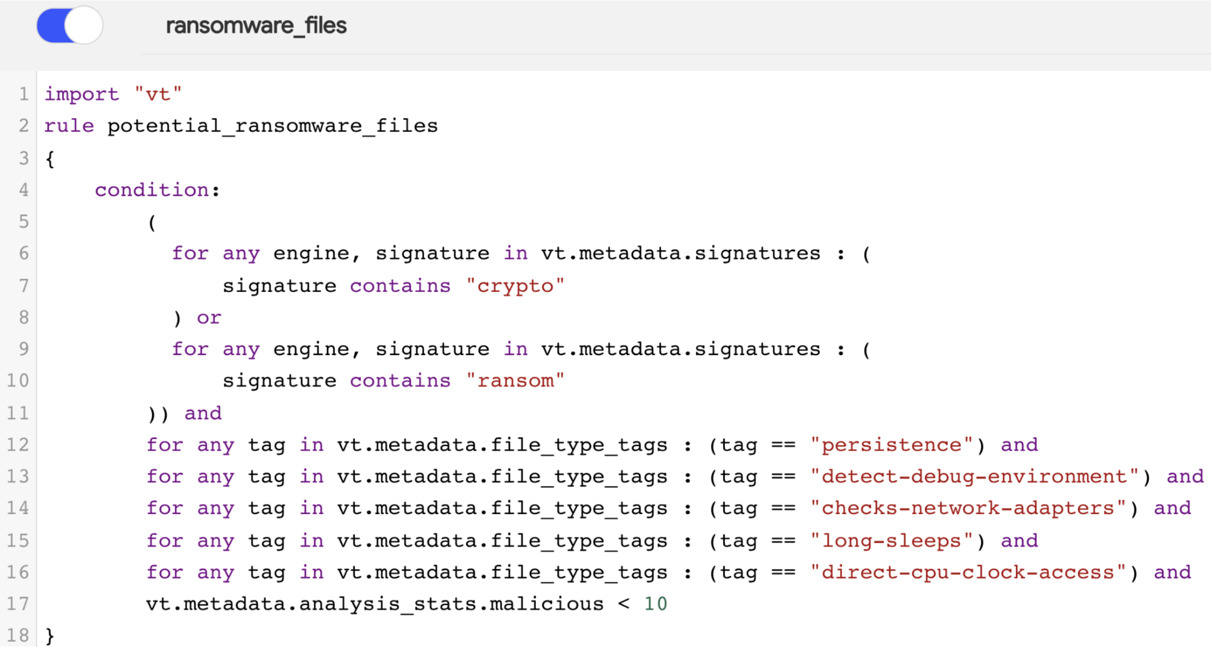
- Searches for files with less than 10 detections: p:10-
- Searches for samples submitted since September 2021: fs:2021-09-01+
- Filters in only those samples that AV vendors or Sandbox providers identify as potential ransom or crypto attacks: (engines:"ransom" or engines:"crypto")
- Takes into account only those tags that are most common among the ransomware samples we have seen in this report: tag:"persistence" and tag:"detect-debug-environment" AND tag:"checks-network-adapters" AND tag:"long-sleeps" AND tag:"direct-cpu-clock-access"

We can focus on files that are potentially exploiting some vulnerability. We can search for them using the “exploit” tag.

"p:10- fs:2021-09-01+ (engines:ransom or engines:crypto) AND tag:exploit"

VTI Search Link

- Advanced Threat Services. VirusTotal extensively uses YARA. Indeed, we developed our own vt YARA module. This allows to easily translate our previous VT Intelligence searches to a YARA rule like the one below:



```
import "vt"
rule potential_ransomware_files
{
    condition:
        (
          for any engine, signature in vt.metadata.signatures : (
              signature contains "crypto"
          ) or
          for any engine, signature in vt.metadata.signatures : (
              signature contains "ransom"
        )) and
        for any tag in vt.metadata.file_type_tags : (tag == "persistence") and
        for any tag in vt.metadata.file_type_tags : (tag == "detect-debug-environment") and
        for any tag in vt.metadata.file_type_tags : (tag == "checks-network-adapters") and
        for any tag in vt.metadata.file_type_tags : (tag == "long-sleeps") and
        for any tag in vt.metadata.file_type_tags : (tag == "direct-cpu-clock-access") and
        vt.metadata.analysis_stats.malicious < 10
}
```

We can find these YARA rules at the end of this post.

To sum up, it is equally important to understand global ransomware trends as to be able to do something about it. In this post we went through different use cases discussing some ideas on how to implement a live cybersecurity threat monitoring system, which can be a game changer for our current security architecture.

At VirusTotal we will keep sharing both our visibility as well as best practices to protect against new attacks and to keep our world a little bit safer. As always, we are happy to hear from you.

Happy hunting!

Appendix - YARA rules

```
import "vt"
rule find_potential_ransomware_files
{
meta:
    description = "Detects potential ransomware related files"
      author = "VT Team"
      reference = "https://blog.virustotal.com/"
     date = "2021-10-04"
      vt_search = "p:10- fs:30+ (engines:ransom or engines:crypto) AND
tag:persistence and tag:detect-debug-environment AND tag:checks-
network-adapters AND tag:long-sleeps AND tag:direct-cpu-clock-
access"
vt_link = "https://www.virustotal.com/gui/search/p%253A10-
%2520fs%253A30%252B%2520(engines%253A%2522ransom%252
2%2520or%2520engines%253A%2522crypto%2522)%2520AND%2
520tag%253A%2522persistence%2522%2520and%2520tag%253A
%2522detect-debug-
environment%2522%2520AND%2520tag%253A%2522checks-
network-adapters%2522%2520AND%2520tag%253A%2522long-
sleeps%2522%2520AND%2520tag%253A%2522direct-cpu-clock-
access%2522/files"
condition:
    (for any engine, signature in vt.metadata.signatures :
         (signature contains "crypto")
         or
        for any engine, signature in vt.metadata.signatures :
         (signature contains "ransom"))
      and
      for any tag in vt.metadata.file_type_tags : (tag == "persistence")
and
      for any tag in vt.metadata.file_type_tags : (tag == "detect-debug-
environment") and
      for any tag in vt.metadata.file_type_tags : (tag == "checks-network-
adapters") and
      for any tag in vt.metadata.file_type_tags : (tag == "long-sleeps")
and
      for any tag in vt.metadata.file_type_tags : (tag == "direct-cpu-clock-
access") and
      vt.metadata.analysis_stats.malicious < 10
}
rule find_potential_ransomware_exploits
{
meta:
    description = "Detects potential ransomware related files using
exploits"
      author = "VT Team"
      reference = "https://blog.virustotal.com/"
     date = "2021-10-04"
      vt_search = "p:10- fs:30+ (engines:ransom or engines:crypto) AND
tag:exploit"
vt_link = "https://www.virustotal.com/gui/search/p%253A10-
%2520fs%253A30%252B%2520(engines%253Aransom%2520or%2
520engines%253Acrypto)%2520AND%2520tag%253Aexploit/files"
condition:
    (for any engine, signature in vt.metadata.signatures :
         (signature contains "crypto")
        or
        for any engine, signature in vt.metadata.signatures :
         (signature contains "ransom"))
      and
      for any tag in vt.metadata.file_type_tags : (tag == "exploit") and
      vt.metadata.analysis_stats.malicious < 10
}
```