TRACKING AN APT VIA SHARED CODE

A Case Study With APT28

Now you can use a malware to hunt a malware. How? By tracing the malware code. Using its ability to trace shared malware code across very large repositories, Cythereal MAGIC connected APT28 samples published by US CYBERCOM to an attack years old attacks: on the Italian Military in 2018 and an attack on TV5 Monde in 2017. All done automatically, without any manual reverse engineering.

July 5, 2019

cythereal
TARGETING TARGETED ATTACKS

Table of Contents

DISCLAIMER

THE DOCUMENT IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. CYTHEREAL INC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENT. THE INFORMATION CONTAINED IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. INFORMATION OBTAINED FROM THIRD PARTY SOURCES IS BELIEVED TO BE RELIABLE, BUT IS IN NO WAY GUARANTEED. SECURITY PRODUCTS, TECHNICAL SERVICES, AND ANY OTHER TECHNICAL DATA REFERENCED IN THIS DOCUMENT (“CONTROLLED ITEMS”) ARE SUBJECT TO U.S. EXPORT CONTROL AND SANCTIONS LAWS, REGULATIONS AND REQUIREMENTS, AND MAY BE SUBJECT TO EXPORT OR IMPORT REGULATIONS IN OTHER COUNTRIES. YOU AGREE TO COMPLY STRICTLY WITH THESE LAWS, REGULATIONS AND REQUIREMENTS, AND ACKNOWLEDGE THAT YOU HAVE THE RESPONSIBILITY TO OBTAIN ANY LICENSES, PERMITS OR OTHER APPROVALS THAT MAY BE REQUIRED IN ORDER FOR YOU TO EXPORT, RE-EXPORT, TRANSFER IN COUNTRY OR IMPORT SUCH CONTROLLED ITEMS.

Tracking an APT via shared code

A CASE STUDY WITH U.S. CYBERCOM PUBLISHED APT28 SAMPLES

1 ABSTRACT

Using just the few malware samples shared by US CYBERCOM, Cythereal MAGIC automatically discovered connections to an attack on Italian Military in 2018, an attack on TV5 Monde in 2017, and to other malware going as far back as 2006. Since these attacks have previously been attributed to APT28, the connection between them is not a new revelation. What makes this analysis significant is that MAGIC traced the connections completely automatically. This study demonstrates that, even in the presence of polymorphism and other obfuscations, MAGIC can track malware evolution with an accuracy comparable to Bindiff, and at scale—across millions of samples—comparable to SSDEEP and Imphash.
Thus, MAGIC can track APTs as they evolve, automatically extract evidence for detection and attribution, and provide threat researchers a quasi-real-time map of the ever changing threat landscape.

2 introduction

Malware, much like any other software, is expensive to engineer. Once perfected malware code is used and reused over a long period of time. Therefore it is no surprise that threat researchers routinely use evidence of similar code between malware as evidence of their common origin. See for example these articles on WannaCry and Lazarus APT, Oil Rig and Green Bug Iranian groups, Flame and Stuxnet, and North Korean malware families.

In this report we use Cythereal MAGIC, Malware Genomic Correlation, to analyze a set of malware shared publicly by Cyber National Mission Force (CNMF), a unit of US Cyber Command. Threat researchers have attributed these CNMF malware to Advanced Persistent Threat group, APT28, (also known as Fancy Bear, Sofacy, Sednit, and Pawn Storm), see Kaspersky and Zone Alarm. While APT28 attacks have been very well studied, the focus of this study is to automatically trace connections from CNMF samples to any recent malware or older malware, such as that reported in a tweet by Florian Roth about finding a new variant of one of the CNMF sample.

> **MAGIC can track APTs as they evolve, automatically extract evidence for detection and attribution, and provide threat researchers a quasi-real-time map of the ever changing threat landscape.**

The eight CNMF malware contain of variants for each of the three tools—Lojack, X-Agent, and X-Tunnel—known to be used by APT28. Lojack, a legitimate laptop theft recovery tool, is said to have been co-opted by APT28 for communication between a victim and C2 servers. X-Agent provides a modular backdoor with keylogging and file exfiltration capabilities. X-Tunnel is a network proxy tool used to create a tunnel between the victims and C2 server and the victim.

Some interesting findings from our analysis are:

- ***MAGIC AUTOMATICALLY TRACED CODE CONNECTIONS FROM LOJACK VARIANTS IN CNMF COLLECTION TO A LARGE NUMBER OF VARIANTS, WITH ONE FIRST SEEN OVER 12 YEARS AGO.***

- ***MAGIC CONNECTED THE X-TUNNEL VARIANTS IN THE COLLECTION TO ONE USED IN AN ATTACK ON TV5 MONDE IN 2017.***

- ***MAGIC TRACED CONNECTIONS FROM THE X-AGENT VARIANTS SHARED BY CNMF TO ONE REPORTEDLY USED IN AN ATTACK ON THE ITALIAN MILITARY IN 2018.***

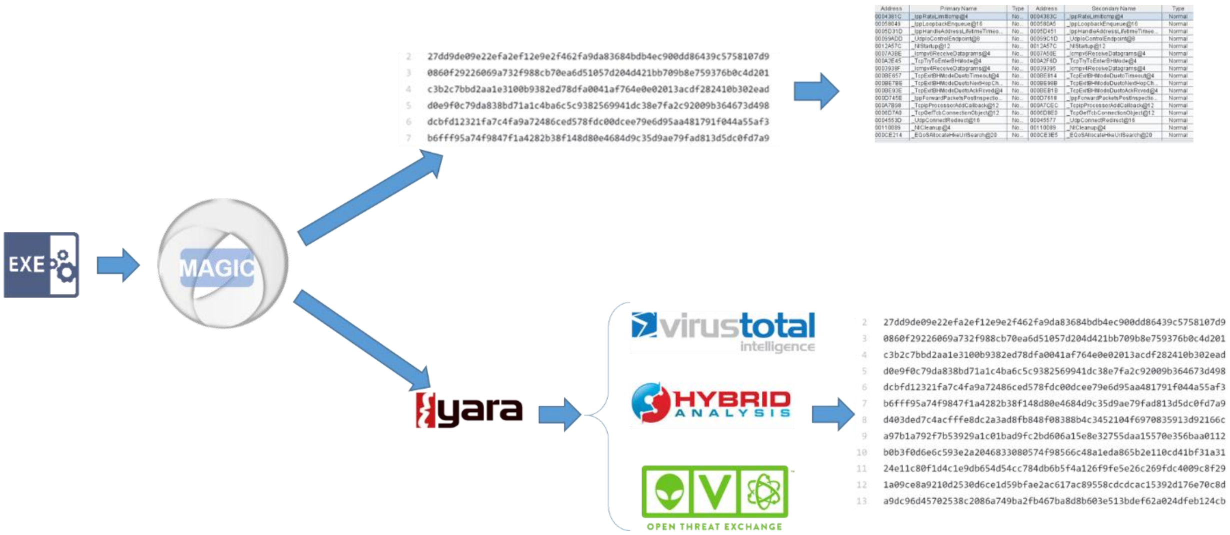
3 WHAT IS MAGIC?

Cythereal MAGIC (Malware Genomic Correlation System) is a “content-based malware retrieval system,” analogous to “content based image retrieval systems”, such as Google Images. It uses an entire malware as a search query to find other similar malware. Though there are systems that use n-grams of mnemonics, PE header data, section entropy, and such properties to create machine learning models for malware detection and malware classification, MAGIC differs from them as follows:

a) **MAGIC is principally a malware search engine for malware with similar code.**
b) **MAGIC uses a much richer abstraction of malware—the code semantics—for creating machine learning models; which enables MAGIC to find similar malware variants with a very high accuracy even when the malware may be obfuscated.**
c) **MAGIC constructs malware signatures for a malware family from one or more malware samples. The signatures, encoded as YARA rules, are constructed from bytecode of malware procedures.**

Figure 1 shows the process of using a malware as a query to search for other malware using MAGIC. There are two work flows. First, find similar malware in the MAGIC database. Second, find similar malware in other threat repositories. For the first, it is just a matter of uploading a malware onto MAGIC, and it does the rest. For the second, MAGIC generates YARA rules, those rules are used to search repositories, such as, Virustotal, Hybrid Analysis, and Alien Vault OTX to find other related samples.

MAGIC significantly improves upon tools currently used for tracing malware variants, such as, Bindiff, SSDEEP, and Imphash. Bindiff, a plug-in to Ida and Ghidra disassemblers, provides very accurate comparison of executables. However, it is best suited for one-off comparisons and does not scale. SSDEEP is designed to find similar images and documents during forensic examinations and is scalable. However, as our analysis shows, it is rather brittle in finding similar malware variants. Imphash is designed to create a quick-and-dirty index of binaries that “import” the same set of libraries. That two executables have the same imphash doesn’t imply that they share code, and executables sharing code need not have the same imphash. At best the imphash matches are good candidates for deeper investigation, with a tool like Bindiff and SSDEEP.



```
27dd9de09e22efa2ef12e9e2f462fa9da83684bdb4ec900dd86439c5758107d9
0860f29226069a732f988cb70ea6d51057d204d421bb709b8e759376b0c4d201
c3b2c7bbd2aa1e3100b9382ed78dfa0041af764e0e02013acdf282410b302ead
d0e9f0c79da838bd71a1c4ba6c5c9382569941dc38e7fa2c92009b364673d498
dcbfd12321fa7c4fa9a72486ced578fdc00dcee79e6d95aa481791f044a55af3
b6fff95a74f9847f1a4282b38f148d80e4684d9c35d9ae79fad813d5dc0fd7a9
```

```
27dd9de09e22efa2ef12e9e2f462fa9da83684bdb4ec900dd86439c5758107d9
0860f29226069a732f988cb70ea6d51057d204d421bb709b8e759376b0c4d201
c3b2c7bbd2aa1e3100b9382ed78dfa0041af764e0e02013acdf282410b302ead
d0e9f0c79da838bd71a1c4ba6c5c9382569941dc38e7fa2c92009b364673d498
dcbfd12321fa7c4fa9a72486ced578fdc00dcee79e6d95aa481791f044a55af3
b6fff95a74f9847f1a4282b38f148d80e4684d9c35d9ae79fad813d5dc0fd7a9
d403ded7c4acfffe8dc2a3ad8fb848f08388b4c3452104f6970835913d92166c
a97b1a792f7b53929a1c01bad9fc2bd606a15e8e32755daa15570e356baa0112
b0b3f0d6e6c593e2a2046833080574f98566c48a1eda865b2e110cd41bf31a31
24e11c80f1d4c1e9db654d54cc784db6b5f4a126f9fe5e26c269fdc4009c8f29
1a09ce8a9210d2530d6ce1d59bfae2ac617ac89558cdcdcac15392d176e70c8d
a9dc96d45702538c2086a749ba2fb467ba8d8b603e513bdef62a024dfeb124cb
```

FIGURE 1 WORKFOW OF USING MAGIC TO TRACE MALWARE CONNECTED VIA SHARED CODE

What is the benefit of finding variants of a malware? It enables threat researchers take advantage of the analyses and IoCs associated with the older malware. For instance, as demonstrated in this report, we found connections to previous attacks through research reports referencing the older malware found by MAGIC. Also, during incident response and for threat hunting, threat researchers often start with a single malware and attempt to find all the IoCs associated to the threat group. In current-state-of-the art this is performed by pivoting between file hashes and IoCs, such as using the relationship graph on VirusTotal. MAGIC automates this process of gathering IoCs by first finding all similar malware and then gathering the IoCs associated to them.

MAGIC significantly improves upon tools currently used for tracing malware variants, such as, Bindiff, SSDEEP, and Imphash.

4 #CNMF MALWARE SAMPLES

Table 1 lists the #CNMF malware samples[1], as identified by their sha256 and sha1, and the dates on which they were uploaded. For ease of reference in the discussion that follows, the #CNMF samples are numbered #1 to #8, shown in column labeled '#'. The numbers assigned have no particular significance.

The column "Matches" gives the number of similar binaries found in the MAGIC database. The column "Component" identifies the groups created by MAGIC. The matches and component grouping are discussed below.

[1] Malware samples shared by CNMF as of July 1, 2019.

TABLE 1 LIST OF CNMF MALWARE SAMPLES AND A SUMMARY OF THEIR ANALYSIS

#	Date	sha256/sha1	Matches	Component
#1	2018-11-05	6d626c7f661b8cc477569e8e89bfe578770fca332beefea1ee49c20def97226e	16	Lojack
		5bc901e9267fa7bb7b14943f5f0299a84a7ef519		
#2	2018-11-05	aa5b25c969234e5c9a8e3aa7aefb9444f2cc95247b5b52ef83bf4a68032980ae	16	
		d578667c9222e7f7835694193576b6554a0bca89		
#3	2018-11-09	dea3a99388e9c962de9ea1008ff35bc2dc66f67a911451e7b501183e360bb95e	5	X-Agent
		7a976e6b79c78d0bdc2140f7a0aab45ccc848c0c		EXE
#4	2019-05-17	b40909ac0b70b7bd82465dfc7761a6b4e0df55b894dd42290e3f72cb4280fa44	0	
		0b28de2c2b0913cc5684461812d294f50fea6105		
#5	2019-01-28	be2e58669dbdec916f7aaaf4d7c55d866c4f38ac290812b10d680d943bb5b757	0	X-Tunnel
		c3212e1e609588cb5736b1fd9aa8581c965ffa08		
#6	2019-01-28	854a522a113b6413ff4db5f0ba0aec98cba3c5ef386311660f6dabab26f6aa14	0	
		43d62b71c7c565622cb69b3af48f8e6431dbfab9		
#7	2019-01-28	e2bea753318d715dfc2f186c49ae3e9c404d0f5df52e959ea546f78a3624bc3b	1	X-Agent
		2d71e7b74d36af33b9481c97bb2b14cfaf2fae6d		DLL
#8	2018-11-09	a5b68575ac4fbe83c23ff991ad0d5389f51a2aef71ee3c2277985c68361cf1cc	1	
		e88768eb8f2d692ad6572a12d115aa78236e8eba		

5 MAGIC MATCHES FOR #CNMF SAMPLES

MAGIC defines similarity based on the percentage of common procedures in a pair of binaries. To be counted as similar, the pair should have at least 70% of the procedures in common[2]. MAGIC compares procedures using their genome. Two procedures between (and within) two binaries are considered same if they have the same genome.

Table 1 shows that MAGIC found 16 matches each for samples #1 and #2, five matches for sample #3, and one each for samples #7 and #8. For the others there were no matches (at 0.70 or greater similarity) found in the MAGIC database, but MAGIC generated YARA rules found matches for these samples, as described later.

In the following subsections we study the matches found by MAGIC. We also use SSDEEP to compare files that MAGIC determines to be similar. As a quick recap, SSDEEP gives a match score between 0 and 100. The higher the number, the stronger the match. A score of 0 implies "no match."

5.1 Matches for #CNMF Sample #1 and #2

[2] MAGIC does provide a method for exploring malware pairs with very low match. That method was not utilized in this study.

Table 2 shows the result of searching MAGIC using the sample #1. In this table and those that follow hashes associated with #CNMF samples are tagged with an asterisk ('*'). The hash of the malware used for query is provided in the first row. The table also shows the "similarity" of the query sample and each matched sample, the "SSDEEP" match between the query sample and the matched sample, the anti-virus "Detection ratio" of the matched sample on Virustotal (as of this writing), the date on which the sample was "First seen" by Virustotal, and a symbol from the "Exports" section of the PE file as a reference.

TABLE 2 MAGIC MATCHES FOR #CNMF SAMPLE #1 AND #2

sha1	Similarity	SSDEEP	Detection Ratio	First seen	Exports
d578667c9222e7f7835694193576b6554a0bca89*	query		49/66	2018-11-05	rpcnetp
2b09346589ccb8c0267d353e90d77d73804ada93	1.00	93	45/70	2019-05-22	rpcnetp
41ad483cc654f662011db550515a9ece5d6a64c5	1.00	91	46/70	2019-05-22	rpcnetp
9093337b3e6611564d72367b6bfb3b3c9c20189d	1.00	91	47/71	2019-05-22	rpcnetp
5bc901e9267fa7bb7b14943f5f0299a84a7ef519*	1.00	99	54/69	2018-11-05	rpcnetp
ef860dca7d7c928b68c4218007fb9069c6e654e9	1.00	96	53/71	2018-09-20	rpcnetp
8e138eecea8e9937a83bffe100d842d6381b6bb1	1.00	91	54/71	2018-09-19	rpcnetp
e8f07caafb23eff83020406c21645d8ed0005ca6	1.00	93	55/72	2018-09-19	rpcnetp
e923ac79046ffa06f67d3f4c567e84a82dd7ff1b	1.00	93	55/72	2018-09-18	rpcnetp
f90ccf57e75923812c2c1da9f56166b36d1482be	1.00	96	53/71	2018-08-09	rpcnetp
fd54f2150cf473c41d93c021e0bbd68d497f0b87	1.00	93	10/70	2018-04-01	rpcnetp
1771e435ba25f9cdfa77168899490d87681f2029	1.00	93	51/67	2017-01-17	rpcnetp
f32db75af71d18007fb6e678c9be1e95dd5b3683	1.00	93	14/69	2008-09-24	rpcnetp
70d1e7efce81c72b87808fdfc3cfcfb1f60bb4fb	0.91	0	4/57	2006-05-15	rpcnetp
9ebd6859925639ce411a7d4ac63896a6a8dad949	0.79	0	0/66	2010-03-30	rpcnetp
12b8b64912b62a64ac988b5838dc32e172e1f16e	0.79	0	1/67	2009-09-28	rpcnetp
87690485148747beb9860996410daab8bbff52a7	0.79	0	0/70	2009-09-07	rpcnetp

The table shows that sample #1 matches sample #2 with a similarity 1.0. Furthermore, a query using sample #2 produces the same result as above, except for swapping of the two #CNMF samples, hence this table serves as a match for both the samples.

Here are some interesting observations:

- There is a perfect match (similarity 1.0) between the query #CNMF file, uploaded on 2018-11-05, and a malware "First seen" almost two years earlier, on 2017-01-17, by Virustotal.
- There is a perfect match with three variants that were first seen on 2019-05-22, that is, about 6 months after #CNMF uploaded the samples.
- There are perfect matches with eight other samples, one of them first seen on 2008-09-24, over 10 years older than the #CNMF samples. Quite surprisingly this old sample has a rather low detection rate (14/69) as recorded from a scan performed very recently, on 2019-06-12[3].

[3] The detection ratio depend on the date on which Virustotal scanned the sample. The scan dates are not included in the tables, and are mentioned in the text where meaningful.

- Sample #1 has very high similarity (0.91) with a file that was first seen on 2006-05-15, almost 13 years before #CNMF shared the samples. Curiously, this file is flagged as malicious by only four scanners when scanned by Virustotal on 2016-11-16.
- As the similarity between the query file and match drops under 0.8, so does the detection ratio. Two such files flagged as benign (detection ration 0) by all scanners on scans done by Virustotal just about a month ago (on 2019-05-27).
- SSDEEP gave a very high match score for all the perfect matches. However, for all the other samples SSDEEP categorically gave a 0 score, implying "no match."
- Each of the files export only one symbol, and that is "rpcnetp". MAGIC doesn't use the strings and symbols in a file for computing similarity. Thus, the existence of the same symbol provides an independent validation that the files found by MAGIC have a common origin.

There is a good explanation for why three of the files reported as similar to sample #1 by MAGIC are flagged as benign, which we present later when discussing the results of YARA search for Lojack.

5.2 Matches for #CNMF Sample #3

Table 3 presents the five matches reported by MAGIC for #CNMF sample #3, posted on 2018-11-09. All of the matched malware were seen almost a year earlier and their detection ratio, as per scan on 2019-06-12, are pretty high.

TABLE 3 MAGIC MATCHES FOR #CNMF SAMPLE #3

sha1	Similarity	SSDEEP	Detection Ratio	First seen	Filenames
7a976e6b79c78d0bdc2140f7a0aab45ccc848c0c*	Query		50/70	2018-11-09	ctinetw; msctray
e19f753e514f6adec8f81bcdefb9117979e69627	1.00	0	48/70	2017-11-03	defupd; AdobeUI
6f0fc0ebba3e4c8b26a69cdf519edf8d1aa2f4bb	0.98	0	46/71	2017-11-07	ctims; ctinet
57d7f3d31c491f8aef4665ca4dd905c3c8a98795	0.93	0	47/69	2017-10-18	ccfm; syclr
2cf6436b99d11d9d1e0c488af518e35162ecbc9c	0.91	0	46/70	2017-11-21	syscheck; sycontr
fec29b4f4dccc59770c65c128dfe4564d7c13d33	0.91	0	46/69	2017-07-13	mmdivx;wsx64

The new, and surprising, insight this table offers is the result of SSDEEP matches. Even for the sample for which MAGIC gives a perfect score, SSDEEP gives 0, implying "no match." And it does the same for other malware even though MAGIC rates their similarity the query sample as over 0.91. These programs do not have any interesting symbols, so the Table compiles some of the filenames associated with each sample. The names do not offer any clue to their common origin, neither do the different names imply a negative.

5.3 Matches for #CNMF Sample #7 and #8

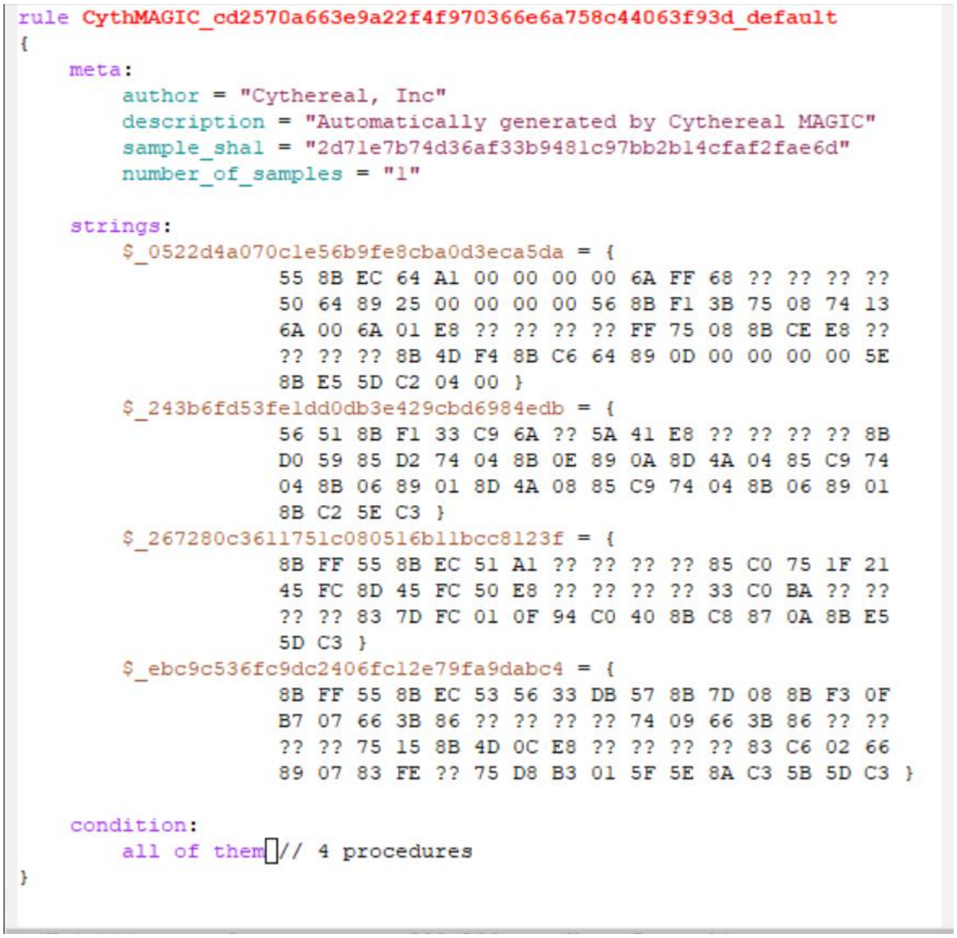
Table 4 gives the MAGIC matches for samples #7 and #8. Turns out these two samples match each other, and do not match with anything else (with a similarity 0.7 or more). Once again SSDEEP gives a match score of 0, implying "no match."

TABLE 4 MAGIC MATCHES FOR #CNMF SAMPLE #7 AND #8

sha1	Similarity	SSDEEP	Detection Ratio	First seen	Filenames
2d71e7b74d36af33b9481c97bb2b14cfaf2fae6d*	Query		52/72	2018-11-09	taskrec; srvcli_
e88768eb8f2d692ad6572a12d115aa78236e8eba*	0.99	0	48/66	2019-01-28	actconv; bmsrv

6 MATCHES FROM MAGIC GENERATED YARA RULES

We now present the result of using MAGIC to generate YARA rules and using those rules to search for matches on Hybrid Analysis. To provide some context on how MAGIC's rules help in tracing malware variants with shared code, we start with providing an overview of how MAGIC generates YARA rules.



```
rule CythMAGIC_cd2570a663e9a22f4f970366e6a758c44063f93d_default
{
    meta:
        author = "Cythereal, Inc"
        description = "Automatically generated by Cythereal MAGIC"
        sample_sha1 = "2d71e7b74d36af33b9481c97bb2b14cfaf2fae6d"
        number_of_samples = "1"

    strings:
        $_0522d4a070c1e56b9fe8cba0d3eca5da = {
                    55 8B EC 64 A1 00 00 00 00 6A FF 68 ?? ?? ?? ??
                    50 64 89 25 00 00 00 00 56 8B F1 3B 75 08 74 13
                    6A 00 6A 01 E8 ?? ?? ?? ?? FF 75 08 8B CE E8 ??
                    ?? ?? ?? 8B 4D F4 8B C6 64 89 0D 00 00 00 00 5E
                    8B E5 5D C2 04 00 }
        $_243b6fd53fe1dd0db3e429cbd6984edb = {
                    56 51 8B F1 33 C9 6A ?? 5A 41 E8 ?? ?? ?? ?? 8B
                    D0 59 85 D2 74 04 8B 0E 89 0A 8D 4A 04 85 C9 74
                    04 8B 06 89 01 8D 4A 08 85 C9 74 04 8B 06 89 01
                    8B C2 5E C3 }
        $_267280c3611751c080516b11bcc8123f = {
                    8B FF 55 8B EC 51 A1 ?? ?? ?? ?? 85 C0 75 1F 21
                    45 FC 8D 45 FC 50 E8 ?? ?? ?? ?? 33 C0 BA ?? ??
                    ?? ?? 83 7D FC 01 0F 94 C0 40 8B C8 87 0A 8B E5
                    5D C3 }
        $_ebc9c536fc9dc2406fc12e79fa9dabc4 = {
                    8B FF 55 8B EC 53 56 33 DB 57 8B 7D 08 8B F3 0F
                    B7 07 66 3B 86 ?? ?? ?? ?? 74 09 66 3B 86 ?? ??
                    ?? ?? 75 15 8B 4D 0C E8 ?? ?? ?? ?? 83 C6 02 66
                    89 07 83 FE ?? 75 D8 B3 01 5F 5E 8A C3 5B 5D C3 }

    condition:
        all of them// 4 procedures
}
```

FIGURE 2 A SAMPLE YARA RULE GENERATED BY MAGIC

6.1 How MAGIC Generates YARA Rules

MAGIC uses the following steps to compute YARA rules from a malware sample:

- **Step 1:** Disassemble a malware binary and split it into functions.

- **Step 2:** Weed out library procedures and procedures that also occur in benign programs.
- **Step 3:** Using a variety of criteria, identify procedures that are good representative of the malware being analyzed.
- **Step 4:** Find other similar procedures in MAGIC's database, and compute an abstract symbolic automata that represents the collection of similar procedures (see paper).
- **Step 5:** Translate the automata into a regular expression, and then generate a hexadecimal string representing that regular expression.
- **Step 6:** Compile all the hex strings for all the representative functions into a YARA rule.

Figure 2 shows a MAGIC generated YARA rule. The rule consists of one or more hexadecimal regular expressions in the "strings" section and a constraint in the "condition" section on how to use the strings for a scan. By default MAGIC generates the constraint "any of them," meaning that a rule succeeds if any of the hexadecimal regular expressions match on the file. Alternatively, if the condition was changed to "all of them" the rule will succeed only when a file has all of the strings.

The rule in Figure 2 is constructed using the sample with sha1 (2d71e7b74d36af33b9481c97bb2b14cfaf2fae6d). Each of the four hexadecimal pattern in the "strings" section is derived from the bytecode of a procedure in this malware. The "condition" section says "any of them," meaning that all of the four procedures should be present in a file for a successful match.

As the number of strings increases, the number of matches for the two type of rules—"any of them" and "all of them"—also start diverging. The "any of them" rules cast a wide net. More strings will catch more malware and also catch more benign. The "all of them" rules are very tight. As the number of strings increases the number of samples that match would decrease. Thus, when the number of procedures are large the rules will catch extremely close variants.

The right condition to use is somewhere between "any of them" and "all of them," and may depend on the specific application.

6.2 MAGIC YARA Rules for #CNMF Samples

MAGIC successfully computed YARA rules for seven of the #CNMF malware. The rules were used to scan the eight #CNMF files to check if they were mutually related. The files that matched the same rules were put in the same group. We next use a YARA rule from each group to search on Hybrid Analysis (HA). To restrict our search to find very close variants we modified the condition in the YARA rules to "all of them." We then Googled the new hashes retrieved from Hybrid Analysis to find threat intelligence reports referencing those malware.

Based on the information found on HA and threat intelligence blogs we determined that each group created from YARA matches corresponds to a component of APT28, as shown in the "Component" column of Table 1. The subsections below describe the result of the YARA searches.

6.2.1 YARA search identifying Lojack Components

MAGIC produced the exact same YARA rule for #CNMF samples #1 and #2. The rule contained 50 procedures. A YARA search on HA matched 32 files, of which 2 were benign. The table below presents the hashes produced by HA, when they were first seen by HA, and the tag assigned to it by HA.

TABLE 5 HYBRID ANALYSIS YARA SEARCH RESULTS FOR LOJACK COMPONENT

Sha256	first seen	Tags
6d626c7f661b8cc477569e8e89bfe578770fca332beefea1ee49c20def97226e*	2018-11-05	LoJack
aa5b25c969234e5c9a8e3aa7aefb9444f2cc95247b5b52ef83bf4a68032980ae*	2018-11-05	LoJack
04567b2462cb60feba5d23f1542bb07fc0be3bb6eee60ea0ba72d0d73ac4649d	2018-10-02	LoJack
eb89d3ee8ce32bd935edc4a1d2e2ad168e239307326abef9fb342da66e38957c	2018-10-02	LoJack
634795a3acbae8964bb31e3ebed7f29208844978a512fc26a8b9a51901f9cab9	2018-09-20	LoJack
eb4e174db15646f71cb1d2c471e5794a8429ca29369c8eff6042122cc6dc6845	2018-09-19	LoJack
a97b1a792f7b53929a1c01bad9fc2bd606a15e8e32755daa15570e356baa0112	2018-09-19	LoJack
500f426f98d4c00d29825f976b9457a274aed781a560a60e89cba4805cd47186	2018-09-19	LoJack
539cdc37c34eebb28a74f0dceeee0331e6ac6f4682e55fddd69d6f9de7ab9b77	2018-08-09	LoJack
06976912957d4c0c7f5d3a478fc8f3dc2ef1057537bc1548554d6569add2ba3d	2018-08-09	LoJack
a6d83fb30af84c18edf829ae4cc29c8c1bfb5eaaf61f9579d2d79c27bd37db59	2018-06-28	LoJack
37f15647c26d475db805048d6592aa153533ac5f4373145c75e24012a51ad9f8	2018-06-28	LoJack
fa8de430fb491d898ee4e557977f036f2aae5f019c3b0552c9e0223da748fc27	2018-02-25	LoJack
65e1f0363f6444783381150ca4914ffcc4599da1262dd4ba0f79cdbdf6f3e4b0	2017-11-20	LoJack
060448ffd71fe2edbb5fe7c6298ad2b077e57fa6ed6d4250fbd799dd85488843	2017-11-05	LoJack
bc99279ab9dc8d6a921e4ecc5432cca954c30c7203407742a034cb43e047ba03	2017-07-24	LoJack
0860f29226069a732f988cb70ea6d51057d204d421bb709b8e759376b0c4d201	2017-07-20	LoJack
dcbfd12321fa7c4fa9a72486ced578fdc00dcee79e6d95aa481791f044a55af3	2017-03-21	Lojax
3f48dbbf86f29e01809550f4272a894ff4b09bd48b0637bd6745db84d2cec2b6	2017-03-21	LoJack
9d566d2d304b7041d2b0524a1420932010b29c46e6849273a2cc4f11f418c94e	2015-10-08	LoJack
a36e05b0f17b43c79a93e7ee9268e7750ceabbe0ac3041200be4f25c5171eef8	2014-03-12	LoJack
0f9ac26dd7f2a6515139afe2933de04e4c6df53394a5bb85c830347fbdcd5c10	2013-09-30	LoJack
3ffdc38b5ed21df52d471cdd8395c853f9b24d641757ce011dfbfd0ebd33b5f6	2013-02-28	LoJack
582396972b015f411db94d4036d3206f453aa8bb9a8c97ee0141da1ddede9674	2012-07-08	LoJack

Since the YARA rules are constructed through bytecode of procedures, it implies that each matching file contains each of the 50 procedures used in the YARA rule for sample #1. Thus, it follows that all of the files found by the YARA search share 50 procedures with sample #1. Couple this with the first seen dates of the matches, and you see that MAGIC has traced a malware that shares code with sample #1 in almost every year going all the way back to 2012, seven years ago.

What does this malware do? We can infer it from the "Tag" assigned by HA. Except one sample, all of the matched samples are variants of Lojack, a legitimate program, described earlier. Since Lojack also has benign used it explains why two of the variants are marked as benign by all of the anti-malware scanners in Table 2.

A Google Search using some of the above hashes led to a repository LOJAX First UEFI rootkit found in the wild, courtesy of the Sednit group, a report by ESET Research, published 2018-09-18. The Repository of IoCs for "Sednit" compiled by ESET Research contains 84 hashes, including most of the hashes listed in Table 2.

6.2.2 YARA search identifying X-Agent EXE Components

MAGIC successfully created a YARA rule for #CNMF sample #3, but not sample #4. The "any of them" YARA rule from sample #3 matched both the samples, which is the reason why the two samples were placed are considered as variants of the same component.

The YARA rule of sample #3 had 92 functions. Table 6 shows the result returned by HA with the "all of them" rule.

TABLE 6 HYBRID ANALYSIS YARA SEARCH RESULTS FOR X-AGENT EXE COMPONENT

sha256	first seen	Tags
dea3a99388e9c962de9ea1008ff35bc2dc66f67a911451e7b501183e360bb95e*	2018-11-09	X-Agent
f2287ddc1376c1ffbf6652d06d115a42e041df1976b321142c0f92dbdb96e82e	2017-11-07	X-Agent
c7661b27a06a3a8c471fbb060ab8cab25fa9546e0a4c5c1101fe8098b2ad11e9	2017-11-03	X-Agent

Even though the rule is highly constrained (requiring all 92 functions to be present), that it still found matches going back a year before #CNMF posted the malware is highly encouraging.

A Google Search on the new hashes discovered led to an AlienVault's OTX Pulse APT28, published 2018-02-25, containing 487 IoCs, almost 9 months before #CNMF shared its sample on 2018-11-09, though the matched sample itself was seen over a year prior on 2017-11-03.

6.2.3 YARA search identifying X-Agent DLL Components

MAGIC produced identical YARA rules for #CNMF samples #7 and #8. The rule selected 27 procedures. Table 7 presents the result from the YARA search on HA using "all of them" version of the rule.

TABLE 7 HYBRID ANALYSIS YARA SEARCH RESULTS FOR X-AGENT DLL COMPONENT

sha256	first seen	Tags
e2bea753318d715dfc2f186c49ae3e9c404d0f5df52e959ea546f78a3624bc3b*	2019-01-28	X-Agent
a5b68575ac4fbe83c23ff991ad0d5389f51a2aef71ee3c2277985c68361cf1cc*	2018-11-09	X-Agent
1228e9066819f115e8b2a6c1b75352566a6a5dc002d9d36a8c5b47758c9f6a45	2018-06-23	X-Agent

The search traced back to a sample first seen about six months prior to the #CNMF upload, a malware said to have been in an attack on Italian Military, dubbed Operation Roman Holiday by Cyblaze published 2018-07-13, and in the Indicators of Compromise for Malware Used by APT28 published by National Cyber Security Center, UK on 2018-10-04.

6.2.4 YARA search identifying X-Tunnel Components

MAGIC successfully generated a YARA rule from payload of sample #5, but not for its original binary. It also generated YARA rule for the original sample #6. A payload is the program resulting from unpacking an original binary. The two YARA rules matched samples #5 and #6, and is the reason these samples are grouped together.

Table 8 presents the result of YARA search on HA using the "all of them" form of the rule from sample #5.

TABLE 8 HYBRID ANALYSIS YARA SEARCH RESULTS FOR X-TUNNEL COMPONENT

sha256	first seen	tags
be2e58669dbdec916f7aaaf4d7c55d866c4f38ac290812b10d680d943bb5b757*	2019-01-28	Dosatyn
86356fa5be88673bcf6f75e9d80d5bfd1a4e8aa621c3565442997e7af3dbded6	2016-12-02	Xtunnel
79f977c8f815c5910df382b920460fd6448103923f4dc128fc56fdf3867c47b1	2016-09-10	Xtunnel

Google search on these hashes led to the report The TV5 Monde Hack and APT28 by Steve McIntyre that presents one of the hashes as a malware used in the attack on TV5 Monde in 2017.

7 SUMMARY

Indicators such as IP addresses, domain names, and malware hashes, currently used to track attacks, are useful over a very short time window as these indicators are transient. However, because the use and reuse of code is dictated by engineering and economic constraints, it follows that malware code can serve as a more stable indicator for connecting and tracing attacks over a longer period of time. Indeed, this is how threat researchers track APTs and high profile attacks, though they currently do it manually. This study shows that through its ability to automatically trace code connections quite accurately in very large malware repositories, Cythereal MAGIC makes it feasible to routinely analyze each and every piece of malware, whether the malware successfully breached the defenses or not. Thus, MAGIC enables threat researchers to track progression of attacks as they evolve and provide a view of the evolving threat landscape that has heretofore not been possible.

Cythereal MAGIC makes it feasible to routinely analyze each and every piece of malware, whether or not the malware successfully breached the defenses.

ABOUT CYTHEREAL

Cythereal is an offensive cybersecurity company that predicts, tracks, and investigates zero-day malware attacks.

CONTACT: For further information contact info@cythereal.com