



[student name]

Final Assessment Report Submission

Case: Imperial Memory

Executive Summary

Provide a brief overview of the case, key findings, the approach taken, and the outcome of the investigation.

The investigation began by examining the Emperor.vmem memory dump file to uncover sensitive data. Through analysis, a password was extracted which successfully decrypted the gift.7z archive. Inside this archive, a file named suspicious.docx was discovered, which appeared to be encrypted or obfuscated.

Upon further inspection using string analysis tools, it was determined that suspicious.docx contained embedded archive data. Using the same decompression tool employed on gift.7z, the .docx file was unzipped, revealing additional files located in a Desktop directory structure. Among these, a file named secret.txt contained clues suggesting that the final step involved hashing a specific file to retrieve the intended answer. This pointed toward a multi-stage analysis process involving memory forensics, archive extraction, file inspection, and hash-based verification.

Findings and Analysis

1. Memory Dump Review (Emperor.vmem)

The investigation began by scanning the memory dump file Emperor.vmem using:

strings Emperor.vmem | grep "gift.7z"

This identified a reference to a protected archive (gift.7z) and revealed a password in plain text memory:

G6Vmc\$Qd5cpM8ee#Ca=x&A3

This finding confirmed that sensitive data such as passwords can be recovered from memory using string analysis — a critical insight for forensic and incident response workflows.

2. Archive Extraction Using 7-Zip

The archive gift.7z was extracted using the 7-Zip command-line utility:

7z x gift.7z

With the recovered password, the archive was successfully decrypted, revealing a file named suspicious.docx. This document was initially assumed to be standard, but further analysis suggested otherwise.

3. Deep File Inspection

Inspection of suspicious.docx using string extraction tools indicated embedded archive content. As .docx files are essentially ZIP containers, it was unzipped with 7-Zip using the same approach:

7z x suspicious.docx

This action exposed an internal directory structure resembling a Desktop environment, containing multiple hidden files. Among them, secret.txt contained hints pointing toward a final challenge involving hashing.

Tools and Technologies Used

Tools:

1. strings (GNU Binutils)

- Used to extract readable text from the memory dump file (Emperor.vmem).
- Helped identify references to the archive gift.7z and locate the password stored in memory.

2. grep

• Used alongside strings to filter and quickly locate specific keywords such as "gift.7z" from a large volume of text output.

3. 7-Zip (7z command-line utility)

- Used to extract the contents of encrypted archives.
- Successfully opened both gift.7z and suspicious.docx, revealing hidden files and data structures.

4. Text viewers (e.g., cat, less, or graphical editors)

• Used to read the contents of files such as secret.txt, which contained clues or instructions for the final stage of analysis.

5. Hashing utility (e.g., sha256sum, md5sum)

• While not explicitly performed in the previous steps, hashing was suggested as a necessary step in analyzing the final file, based on the contents of secret.txt.

Technologies Involved:

1. Memory Forensics

• Leveraged volatile memory analysis to uncover sensitive data like passwords and file references, demonstrating the risks of storing credentials in plaintext RAM.

2. File Compression & Archiving

- Involved encrypted archive formats like .7z and .docx (a ZIP-based container).
- Showed how data can be hidden or nested within multiple layers to evade detection.

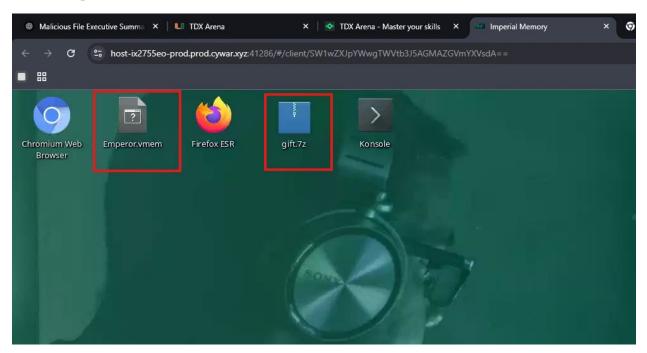
3. Command-line Interfaces (CLI)

• All major operations were conducted via terminal commands, showcasing the power and speed of CLI tools in forensic workflows.

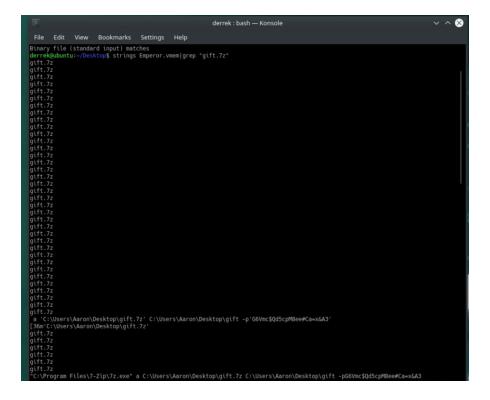
4. Data Obfuscation Techniques

• The use of a .docx file to hide additional archives and files reflects common tactics used in malware and digital forensics challenges to bypass casual inspection.

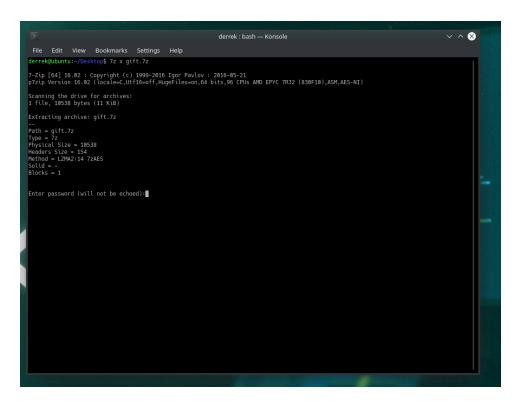
Investigation Process



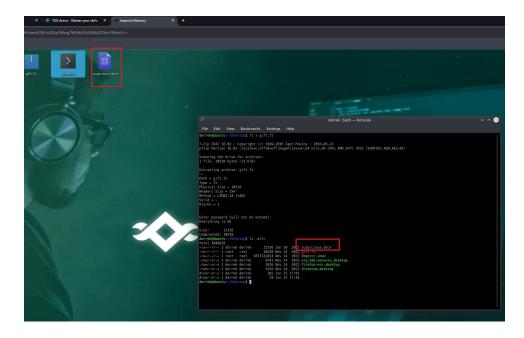
1 These are the files that needed to be investigated



2 This is the Information extracted form Emperor.vmem strings Emperor.vmem | grep "gift.7z"



3 7zip was used to unzip the gift.7z that needed the password extracted in step 2 7z x gift.7z



4 a suspicious.docx file was placed in the Desktop

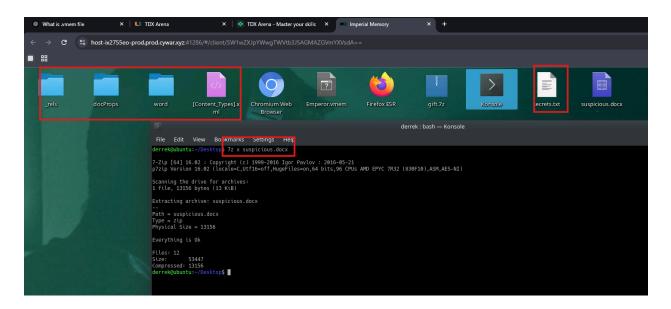
$\overline{}$								dei	rrek : bash —	Konsole							\sim \sim \otimes
File	Edit	View	Bookma	irks	Settings	Help											1
derrek@ P!n@lZ[l"%3003	Content ۲ښ¢۵۷Б	· <mark>∼/Desk</mark> _Types	<mark>top</mark> \$ cat].xml Ø(8888n6 7+888-1 1088	icious.d 0E 00000 d& 00 00A0	Jb 祛*0>0- 601400L60	-R 6 {V 00 400 9#000S		₩?0PW00tt4Q+	ชิชิ"ซิงเลชิชิชิ T	1. บติติติ	ā ∩லிலிபல்≲லில்	ນຍີ–⊓∕ໜິດ	DOXOSO (DO		ŵNŵI 2ŵFŵ	RR 6100-
Fk	նինյն	00 ښ000	0q0100?6				0m0000P!0000	N			19000	9,11000000000	9-07 993	997929199	ia<∟aa7aa‱;		.aa. a@a-
00	ÛÛ^Û[(0000×0000		ிரில்லி#⊤) (RVRR)	<u>rren</u> re		rels/.r	els 0(0000j		∭©N/c6	0000[[l00]01	00<00]0a	GÛÛäÛZSÛ	Fu 00]0∭0 U		
					/] 0 100@0	00000 00		88qU886	~300100jH[{	9=E0000-							
f?003-0 20000z0		20))0, 0	10/%00bi 0/0		00?60!Y0	_©o]A©©©P		ord (doc	ument.xml00	ກະຕິດຕໍ່ຕໍ່ຕໍ່ຕໍ່ຕໍ່ຕໍ່	யல்	ha4 ≂® \7!	"ATA188"	1761-060	c©µ≏Xf©T)©0	ரில்ல் வி	-^^+-\ ®ml
ØG8Ø9]Ø	3q 88888	/0000d0	h © -@®*<				w	01 07 000	ument.xmtøø	augogggggg g`	j# 9	11a4-2 9 /2	010300	7 1 6 : 7060	-@XTØT/00	91000 00	
000ńb<0 gci0Lj0	00000021 000MXV0	000&0H0 ;	00QG0!0/						80>') 800000 	ii:00000q\$00	00#0}1	}0 0w0X00sA	∖sp©©dX©i	¢a ûû WX ûû	4K©©NPcp©0	10PQ	
. X#200×	.eeeenei	96D:6(6	GGGYEARG	Ø ØTCØ	1000005100	ig+øsøøøø	9000=0m0h.))	6666666		f©Ân ©©©©©b©∺	⊧=v iûiû -	-ŵcŵ%`ŵ<:ŵ	ûû\$Xû'8û	ŵ *X0 ©© ©1	Xûwerrord	û∨ rîr 8≈46 r	00000>0}h
	000-0 20)0pi00	MOOYOOM	6 6666	0-00!Fi	0=000P!0	d©Q@word/		ment.xm ŮVw©5©j	000000000000 l.rels 0(00	05[00Do0;01 00j00E000003]	r@lv-@ }-;}PB	0000cXP00t(B0lj!000E?(0E000c09 0,	0r00r0L0 00`HI00`(000- -+∲ ∨00] 00r00sπ600	ÛPQH 00+0 00~00{000	;0ug))0u10 时存{0*x0
! 00g006	word/1	theme/t	heme1.xm	n lêykê	00000z00	a050H0k06	5 _000 Rk 000 i			p:th aaaa onv[DAAEAVAAAC7	207,0000	02000]00	5015حم8.00	x00+00000	1(:0>2000
									ØØB	1000c0100013	കുറക	6 6611766666		e.ce			
										1000/0100013	o-@u]@	0 000:00000	00045000		q 00 \$aá}0,0	∖GH ©® e	00
			Foééocé			444-1 744	-	a a. a a	1.00	T-CD-404							,
,			, F000960					ievenwae	@0_"\$ Q00		ØqaØ1;	;>00000P\$\$L1	t∵ 0s0;0 k	!* dsrC©©	Ŵ[8
005-000	5 000^00 °	Fn ®® Ac®	X004∰0NS	5.00f-0	800(mZt0	0020000 <u>0</u> 0	¢]_} ¢¢Mo										
0(10[06	}∂∂@ ₂ !F/	[@m@kKx	® *®®+® 0d	106666	Ch © "IG.f		?0										
0000000	- dr+0000					%0080	0\$0 0f(a0+0	*R00000	188(:8QN:388	100#Nf00^0n(80ŋ000	00 000Ĭ_>0t	10000e00	y07?}000,	/ 666 000×0ø	000000_00	A00g00?{
000000		≩:>vn¢∰	6hY 000 80	ØØKtØP													
67)0686	90Qsl000	E0.!0]0	Ì306≈K0"	Vôôc#((00z40006 04m008=Fm		106\006	@0400E2000_0	90ap!00!0000	96 0 ані	i 000=00 4\$00	djC 0"000	>00gV0\$0	ŵû +b		
X								d	errek : bash	— Konsole							\sim \sim \otimes
File	Edit	View	Bookr	marks	Settin	as Heli	n										
@#[8	Luit	1644	BOOKI	TIALKS	Securi	gs nei	٢										
PQDazF	F																
(\$]a																	
2cAd 2=+4(<	<r< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></r<>																
+'>bY																	
#9:2 docPro	ops/cor	e.xml															
P2G\$	000/001																
SGq' u`i{u																	
*(2)	}d																
>{ Ź/0		vml															
zG3k	ops/app	.xmt															
WT99																	
{oDB H4TV																	

T[=/		
Er83		
3V@h		
)<6!		
`0Ix2]:		
qzy;A		
=Z"L		
Urs6		
j#Mo		
VaRU		
{ :HT		
[Content_Types].xmlPK		
_rels/.relsPK		
word/document.xmlPK		
word/_rels/document.xml.relsPK		
word/theme/theme1.xmlPK		
word/settings.xmlPK		
word/styles.xmlPK		
word/webSettings.xmlPK word/fontTable.xmlPK		
docProps/core.xmlPK		
docProps/app.xmlPK		
H4TV		
secrets.txt		
derrek@ubuntu:~/Desktop\$		
der rengasarrent / besit opp		

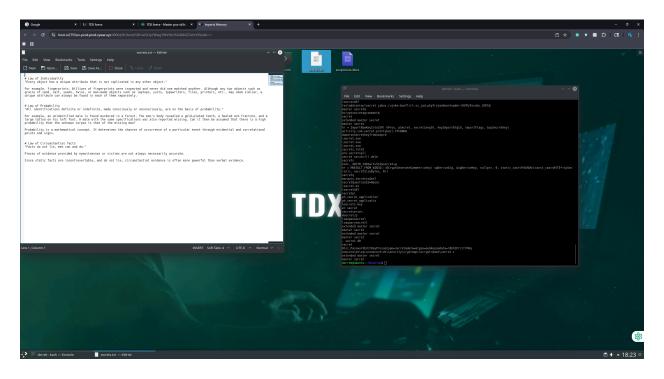
5 After inspecting the file it was found using

strings suspicious.docx

that the document had embedded archive data.



6 after using 7z to decompress file, three folders and two files were placed in the Desktop



7 After reading secrets.txt it was implied that everything had a unique serial or code selecting this file provide a md5 hash by using

md5sum secrets.txt

Recommendations

Based on the findings from the memory dump and file analysis, it is strongly recommended to implement stricter controls over sensitive data retention in memory. The fact that a password used to access an encrypted archive (gift.7z) was found in plain text within the Emperor.vmem file highlights the critical need for secure memory handling. Developers and system administrators should ensure that applications handling sensitive information clear memory buffers immediately after use. Additionally, endpoint protection tools should be configured to regularly scan volatile memory and flag the presence of plaintext credentials or unauthorized archive files.

From a security operations perspective, it is advisable to deploy automated tools that continuously monitor endpoints for suspicious file structures, such as encrypted .docx files containing hidden data or unusual archive nesting. Endpoint Detection and Response (EDR) systems, paired with scheduled memory analysis (e.g., via Volatility), can help detect and investigate threats that use advanced file obfuscation techniques. Lastly, hashing and integrity checks should be integrated into the incident response process to verify file tampering and to support forensic timelines during post-incident analysis.