

작성자	허은정
분석 일자	2024.05.09
작성 일자	2024.05.09
분석 대상	Hello_SuNiNaTaS.pdf
문서 버전	3.0
작성자 E-mail	dmswjd4315@yonsei.ac.kr

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3

4. Write-Up.....4


5. Flag.....9

6. 별도 첨부 10

7. Reference 11



1. 문제

URL	http://suninatas.com/challenge/web31/web31.asp
문제 내용	<p>*안내: 본 PDF 파일은 PC에 유해한 작업을 하지 않습니다. 단순 문제 풀이용입니다. 악성코드가 첨부된 PDF를 분석하여 Flag를 찾으세요.</p> <p>인증키 형식: lowercase(MD5(Flag))</p>
문제 파일	 Hello_SuNiNaTaS. pdf
문제 유형	disk forensics
난이도	1/3

2. 분석 도구

도구명	다운로드 링크	Version
PDF stream Dumper	http://sandsprite.com/blogs/index.php?uid=7&pid=57	v0.9.624

3. 환경

OS
Window 11 64-bit

4. Write-Up

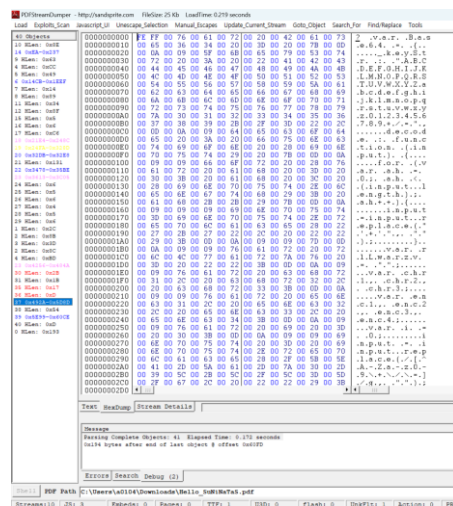
파일명	Hello_SuNiNaTaS.pdf
용량	24.6KB (25,232 바이트)
SHA256	d1d3fd81952ffab1d52509a0d6dd7bcd27017e082ec99d4b0c4a0004577c4fdf
Timestamp	2024-05-09 16:06:49

문제를 보면 PDF 가 존재하여 다운로드 받았다.



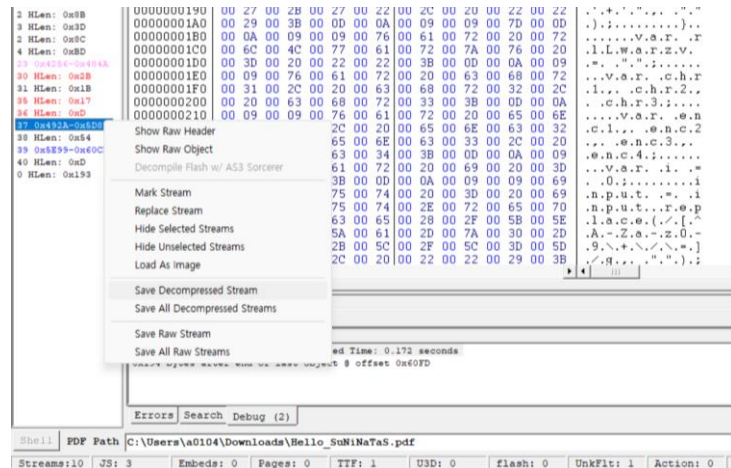
[사진 1] 문제 PDF 내용

문제파일을 다운 받은 후 열면 [사진 1]과 같은 내용이 나오고 다른 문제점을 찾지 못하였다. 이를 봤을 때 pdf를 분석할 프로그램이 필요할 것이라 생각하여 pdf Steam Dumper를 이용하였다.

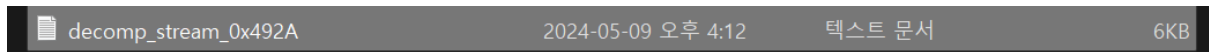


[사진 2] pdf Stream Dumper을 활용하여 본 Hello_SuNiNaTas.pdf

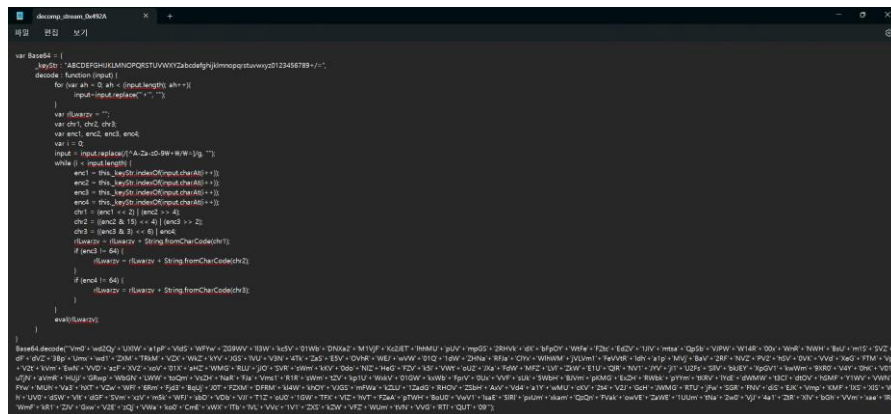
Pdf Steam Dumper을 활용하여 Hello_SuNiNaTas를 본 결과, 37번째 object에서 자바스크립트 코드가 있다는 것을 확인할 수 있었다.



[사진 3] 37번 object Save Decompressed Steam



[사진 4] 추출한 txt문서



[사진 5] 추출한 txt 문서 내용

37번째 object를 Save Decompressed Steam 한 후, 파일을 메모장에서 열어본 결과 위의 코드 부분은 base64 디코딩 코드를 확인하였고

VmOwd2QyUXlWa1pPVldSWFYwZG9WVlI3Wkc5V01WbDNXa2M1vjFKc2JETlhhMUpUvmpGS2RHVkdXbFp0YwTfFeFZtceDZV1JlVmtsaQpSbVJPWW14R00xWnRNWHBsUm1SSVZtdFdVZ3BpUmxwd1ZXMT RkMVZXWkZkYVJGS1VUV3N4TkZaSE5V0VhRWEJwVWO1Q1dWZHNaRFJaC1YxWlhwMjVlVlm1FeVvTR l dha1pMVjBaV2RFNVZPV2hSV0VKVvdXeGFTMVpXV2tkVmEwNVVDazFXV2xoV01XaHZWMGRLUjJ0 SVRrSWMkkV0doNlZHeGFZVk5lVWtoU2JXaFdWMFZLVlZkWE1UQlRNV1JYVjJlU2FsSlVbkJEYXp GV1kwWm9XROV4YOhKV01uTjNaVmRHUjJGRwpWbGNLWWtoQmVsZHNaRFJaVms1R1RsWmtZVkp1UW xkV01GWkxWbFprVOUxVVFsuK5WbHBjVmpkMGExZHRWbkpYYmtKRVIYcEdWMWt3CldtOVhSMFY1W VVWNFYwMUhVa3hXTVZWwF16RmFjd3BqUjJOTfZXMDFRMk14Wkh0YVJGSmFWakZLU1ZadGRHOVZS bHAXVvd4a1YwMUckV2t4V2JGcHJWMGRTUjFwSGRFNVd1SEJKVmpkMF1XSXlSWGhUV0dSWVlt dGF SVmxzVm5kWFJsbDVBbVJlT1Z0U01GWTFXVlZhVTFZeApTWHBoU0VwV1lsaE5lRlpxUmxkamQzQn

[WHS-2] .iso

FVakowVEZaWE1UUmTNa2w0VjJ4a1ZtRXlVbGhVVMxaeIRrWmFkR1ZJVGxwV2EzQjVWako0CmExW
XITbIVLVVc1V1ZXSkZWVZWUmtVNVVGRTlQUT09

해당 문자열을 Base64 디코딩해 주는 사이트(<https://www.base64decode.org/ko/>)에서 돌려보았다.

Decode from Base64 format

Simply enter your data then push the decode button.

Vmr0w2QyLXWlaTpPvIdSWFYwZG9Ww03Wkc5V01WbDN0a2M1VjFKc2JlTlhlMjUvVmpGSGZRHVkd06FpOYWFfZdcEdZV1JlVmtaQp8bVJPWW14R00wWhR
NWbHsJm1SSVZuF4VZ3BpUmwed1ZMTRMwVZXWkZVYJG8VUV3M4TKZaSE5VOVhRWEJaW01Q1dWZHNarFJaCYvWhWMVLm1FvVVRdha1pMVJBa
V2RFRNFZP2hSV0KVVv0aGFTMVpXVZ8VnEwNVDaZFXVzeV01XaIZWMGRLUjOSVRSaWmkKV0doN6ZheGFZv5VYWh0J2JXafWMFZLVZWE1UQRNV
1JYVj1UZF5SIVbUJEYXpGV1kwWm0XRV4Y0hKV01uTjNaVmrRHUJGRwpW6GNLWb0QnVaZHNarFJaVms1R1RaWmtZVp1UWkaV01GWkxWfprV0uXVfS
Uk8WbHbVmsKAGEZHRWapY1mKRYE0EMW0C3aD7hSMFY1WVWNPYmUhh3dXTVZwWfR6RmFjd38uJjUJTFZAMQFRMk4WkoVYJGSmFwaZ
LU1Za6GRHOVZ8hAaVv04a1YwMUKV2H4VZGdUWMGRTUJfW6GRFNVa8EJKVmpkMFIXSXSISWGHUV0dSWb9aGFSvmozVmsWfJaaZVD6vJIT1ZaU01G
WTFXVzhVTFZaPThB0UUVwV1kaESRlpUmkamQzQnFVakowVEZaWE1UUmTNa2w0VjJ4a1ZtRXlVbGhVVMxaeIRrWmFkR1ZJVGxwV2EzQjVWako0CmExW
XITbIVLVVc1V1ZXSkZWVZWUmtVNVVGRTlQUT09

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8

Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☒ Live mode OFF

Decodes in real-time as you type or paste (supports only the UTF-8 character set).

←

DECODE

→

Decodes your data into the area below.

Vmr0w2QyLXWlaTpPvIdSWFYwZG9Ww03Wkc5V1JbDNXa1JTVFKdGVGMZNaEXvmpGYWRHv6
RmrROYmcGM1ZHUxpRmrRvmtWUgplRpwVW14d1VWZFdaRFJUTWxNFZHNuXQXBPum5CWRVdsZDRZ
V1ZXVZ5KvEmEYUmfWakZLVZwWkE5VOWhRWEJUVWaaS1ZWwkd0a0Uck1WWhWMhVv0aR2NtIZI
WGR6VgaYvNH4h3SWWhV0KVV0dMTBMRXVZSSaJUVnBdaZFWY0Z3WGEchJWMhN3ZV6GR2FG
VoaYNaBhaZDRZV6GCTZKXVJuc3aWwFZLVZWE1UQRNVpHjUha1dVnUkXaJEYXpGV1k
Wm0XRV5YUV4V01HJkxWMpXyZFacwpR2LW01Q2hZHNarFJaVFKSVZdG9VRp1UWkaV01G
WkxWfprV0dSR1pHE5WbHBJUJY0YWYDhTWGRYmLaRVaVndXRIScRHOVhSMFY1WVWaaU1Yx
SXpSEpWYhNafZqRidj3BqJjUJTFZAMQFRMk4V2aKvMEYUJhUVZrTKZaGvITpWabSVJ4
a1YyShUKUW5WVWVJTVVR8U5JFE9PQ==

[사진 6] Base64 디코딩

Decode from Base64 format

Simply enter your data then push the decode button.

SSh8eSBzb3JyeSwgVGhpcyBpcyBub3QgSVZ5Eh:

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8

Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☒ Live mode OFF

Decodes in real-time as you type or paste (supports only the UTF-8 character set).

←

DECODE

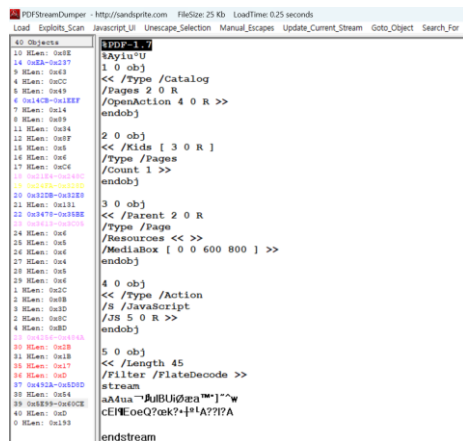
→

Decodes your data into the area below.

I am sorry. This is not Key-It!

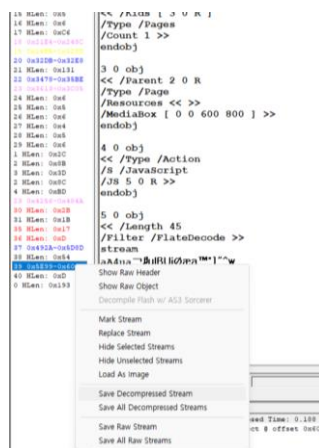
[사진 7] Base64 디코딩 결과

Base64를 계속 복호화 시도해본 결과 다음과 같은 문구가 나오게 되었고 이를 통해 이것은 key 값이 아니라는 것을 알게 되었다. 따라서, 다시 PDFStreamDumper로 돌아가보았다.

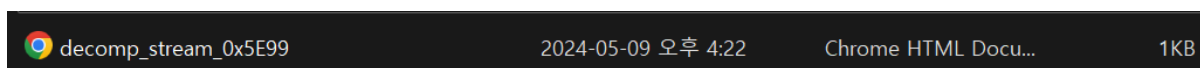


[사진 8] 39 object 내용

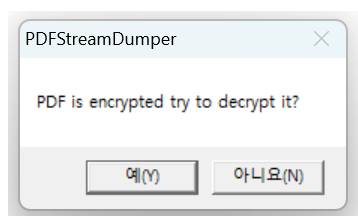
39번째 object에서 PDF 헤더를 확인할 수 있었고 이를 통해 PDF 안에 PDF가 하나 더 있다는 것을 알게 되었다.



[사진 9] 39 object Save Decompressed Stream



[사진 10] 추출한 PDF



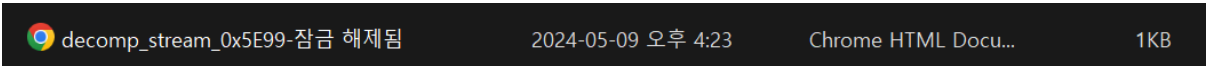
[사진 11] 추출한 PDF를 PDF Steam Dumper에 연 사진

39번째 object도 save Decompressed Stream을 한 후, PDF Stream Dumper로 열어보았더니 PDF가 잠겨 있다는 것을 알 수 있습니다.

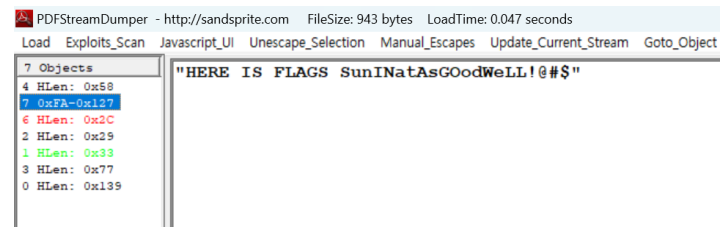
[WHS-2] .iso



[사진 12] PDF 잠금 푼 사이트(<https://smallpdf.com/kr/unlock-pdf>)



[사진 13] 잠금 풀린 PDF



[사진 14] 해당 PDF를 PDF Stream Dumper로 연 내용

[사진12]를 통해 잠긴 PDF 파일을 풀고, 다시 PDF Stream Dumper를 이용하여 확인해보았다. 해당 PDF의 7번째 object에서 Flag 값을 찾을 수 있었다.



[사진 15] flag값을 MD5시킨 결과

해당 flag값인 SunINatAsGoodWeLL!@#\$을 [사진15]에서 MD5 인코딩 시켜보면 13d45a1e25471e72d2acc46f8ec46e95라는 auth key가 나오게 됩니다.

5. Flag

13d45a1e25471e72d2acc46f8ec46e95

6. 별도 첨부

7. Reference