[TECHNICAL TEARDOWN: EXPLOIT & MALWARE IN .HWP FILES]

Share this...

This article will focus on teaching analysts on analysing malicious JavaScript code within the HWP files and a walkthrough of how we can analyse .HWP files that was used to deliver malware.

[1st Sample used in the analysis]
MD5: 8EB5A3F38EB3DE734037AA463ADE7665
SHA256 [illegible]67EE
As of w

[Part 1
We nee
Inside t use
SSView ous
.hwp fil

Most of below
that the

Howeve ll,
think of "**DefaultJScript**" as VBA in Office documents.

1

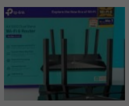
[Part 2 : Getting Started]
For those who want to follow along. Do note, this is a MALICIOUS file, so please do the analysis in a "**safe**" environment.

Now, let's start getting our hands dirty…and open the suspicious .hwp file with Cerbero Profiler.

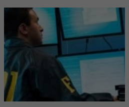
As we can see from the image below, the data within "**DefaultJScript**" looks gibberish. So how do we make sense out of it?

LATEST VIDEOS

TUNNELCRACK: TWO SERIOUS VULNERABILITIES IN VPNS DISCOVERED, HAD BEEN DORMANT SINCE 1996


HOW TO EASILY HACK TP-LINK ARCHER AX21 WI-FI ROUTER

US GOVT WANTS NEW LABEL ON SECURE IOT DEVICES OR WANTS TO DISCOURAGE USE OF CHINESE IOT GADGETS


24,649,096,027 (24.65 BILLION) ACCOUNT USERNAMES AND PASSWORDS HAVE BEEN LEAKED BY CYBER CRIMINALS TILL NOW IN 2022

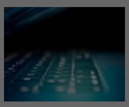
HOW CHINESE APT HACKERS STOLE LOCKHEED MARTIN F-35 FIGHTER PLANE TO DEVELOP ITS OWN J-20 STEALTH FIGHTER AIRCRAFT [VIDEO]

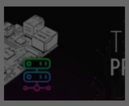
VIEW ALL

VULNERABILITIES

MASSIVE NVIDIA GPU EXPLOIT FOUND. HOW HACKERS CAN TAKE DOWN 35% OF AI SYSTEMS IN CLOUD!

BLAST-RADIUS ATTACK EXPLOTING CRITICAL RADIUS FLAW COULD COMPROMISE YOUR NETWORK


14 MILLION SERVERS VULNERABLE TO CRITICAL OPENSSH BUG: BECOME REMOTE ADMIN WITH CVE-2024-6387

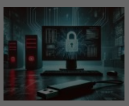

HOW SAFE IS YOUR TINYPROXY? STEP-BY-STEP GUIDE TO EXPLOITING TINYPROXY'S ZERO DAY VULNERABILITY

ETERNAL MALWARE: CVE-2024-3400 ROOTKITS PERSIST THROUGH PALO ALTO FIREWALLS UPDATES AND RESETS

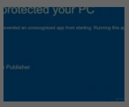
VIEW ALL

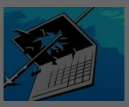
TUTORIALS

5 TECHNIQUES HACKERS USE TO JAILBREAK CHATGPT, GEMINI, AND COPILOT AI SYSTEMS


THIS HACKER TOOLKIT CAN BREACH ANY AIR-GAPPED SYSTEM – HERE'S HOW IT WORKS

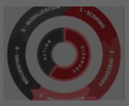
HACKING PAGERS TO EXPLOSIONS: ISRAEL'S COVERT CYBER-PHYSICAL SABOTAGE OPERATION AGAINST HEZBOLLAH!

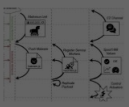

FIVE TECHNIQUES FOR BYPASSING MICROSOFT SMARTSCREEN AND SMART APP CONTROL (SAC) TO RUN MALWARE IN WINDOWS


HOW MILLIONS OF PHISHING EMAILS WERE SENT FROM TRUSTED DOMAINS: ECHOSPOOFING EXPLAINED

HOW TO IMPLEMENT PRINCIPLE OF LEAST PRIVILEGE(CLOUD SECURITY) IN AWS, AZURE, AND GCP CLOUD

THE 11 ESSENTIAL FALCO CLOUD SECURITY RULES FOR SECURING CONTAINERIZED APPLICATIONS AT NO COST

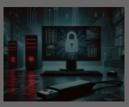

HACK-PROOF YOUR CLOUD: THE STEP-BY-STEP CONTINUOUS THREAT EXPOSURE MANAGEMENT CTEM STRATEGY FOR AWS & AZURE


WEB-BASED PLC MALWARE: A NEW TECHNIQUE TO HACK INDUSTRIAL CONTROL SYSTEMS

THE API SECURITY CHECKLIST: 10 STRATEGIES TO KEEP API INTEGRATIONS SECURE

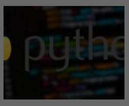
VIEW ALL

MALWARE


THIS HACKER TOOLKIT CAN BREACH ANY AIR-GAPPED SYSTEM – HERE'S HOW IT WORKS

HACKERS' GUIDE TO ROGUE VM DEPLOYMENT: LESSONS FROM THE MITRE HACK

ETERNAL MALWARE: CVE-2024-3400 ROOTKITS PERSIST THROUGH PALO ALTO FIREWALLS UPDATES AND RESETS


MAJOR PYTHON INFRASTRUCTURE BREACH – OVER 170K USERS COMPROMISED. HOW SAFE IS YOUR CODE?

HOW TO EXPLOIT WINDOWS DEFENDER ANTIVIRUS TO INFECT A DEVICE WITH MALWARE

VIEW ALL

2

As i've mentioned earlier, most of the streams within .hwp are "**zlib compressed**"
So let's "**Select All**" within the "**DefaultJScript**" stream and press "**Ctrl+T**"

Now let's add "**Unpack Zlib**" and remember to check the "Raw" checkbox and add it as shown in the image below.

3

Then let's press "**Preview**" and have a look.

4

After de seems
to be in

Now let's add in another filter to remove the "00" bytes.
Select "**Replace**", change the mode to "**Bytes**" and add in "**00**" for the "**In**" value as shown below.

5

We should get back something like the one shown below.

6

If we were to analysed the decoded JavaScript, we can see more interesting stuff as shown in the image below.

So it seems that the JavaScript is doing Base64 decoding of the very long string and dropping it as "msvcr.exe"
I wrote the following Ruby script to decode the Base64 String.

```
require "base64"
content = File.read('file.b64')
decode_base64_content = Base64.decode64(content)
File.open("Output.exe", "wb") do |f|
  f.write(decode_base64_content)
end
```

After Base64 decoding the string, the output file looks like this,

The hash of this malware is
765834b1b780dacda8baa671c76328445cb8278099bad375ee22130f48920a7a
We wor

[2nd S
MD5: **a**
SHA256
As of w

[Part 3
This is a 0808).
I drew a exploit
works.

For this particular exploit, the first thing we should be looking at is **BinData/BIN0001.EPS** as shown below.

There is an unknown error upon opening the document using hwp2010.

Neverth
Let's do ee that
the exploit was indeed executed and connect to
www.ethanpublishing[.]com/ethanpublishing/phpcms/templates/default/member/account_manage/teacup.jpg
if we use FakeNet or similar tools.

We suppose that **teacup.jpg**" is most likely the payload. However, the jpg file is no longer found using the url so we cannot conduct further analysis on it.

Let's go on to focus our analysis on the vulnerablity that was exploited by the eps file.

Opening the file eps file in the text editor we can identify a few components of the exploit.
The green block represents a NOP sled using **0xB5**.
The blue block represents a NOP sled using **0x90**.
The red block represents the shellcode.

Following the shellcode is this line of post script command

```
500{/A1 65535 string dup 0 D40 putinterval def A1}repeat}
```

This command would execute a "**Heap spray**". 500 blocks of the NOP sleds and shellcodes would be '**sprayed**' in the memory. The NOP sleds and shellcodes is allocated as a string with a length of 65535 characters.

Next we want to determine which vulnerable process is the exploit targetting.
We do so by trying to search for traces of the NOP sleds and shellcodes in the memory of the vulnerable process.
At first it looks like the vulnerable process is likely hwp.exe or HimTrayIcon.exe

Howeve ses.

At this child process

One 'tri " which i ellcode to run c

Now we sleds and she

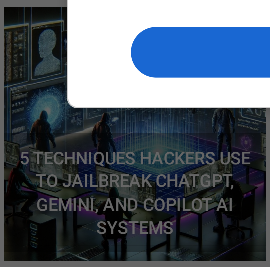
Now after locating the vulnerable process, we have to debug into it to locate where the vulnerable code is exploited.
We now located the code in where hwp.exe created the gbb.exe process.

We shall modify the "**CreationFlags**" to CREATE_SUSPENDED. This would allow us to attach debugger at the start of the execution of the gbb.exe process.

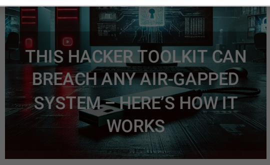
After tracing the code we located the instructions in gsdll32.dll that executed the NOP sled "**0xB5B5**" which is MOV CH,B5

From the vulnerable instructions, we can more or less conclude that the vulnerablity is indeed based on **CVE-2013-0808**

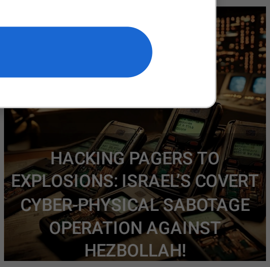
Source:https://www.vxsecurity.sg



5 TECHNIQUES HACKERS USE TO JAILBREAK CHATGPT, GEMINI, AND COPILOT AI SYSTEMS



THIS HACKER TOOLKIT CAN BREACH ANY AIR-GAPPED SYSTEM – HERE'S HOW IT WORKS



HACKING PAGERS TO EXPLOSIONS: ISRAEL'S COVERT CYBER-PHYSICAL SABOTAGE OPERATION AGAINST HEZBOLLAH!

Alisa Esage G

Working as a cyber security solutions architect, Alisa focuses on application and network security. Before joining us she held a cyber security researcher positions within a variety of cyber security start-ups. She also experience in different industry domains like finance, healthcare and consumer products.

ON: NOVEMBER 23, 2016 / IN: IMPORTANT, INCIDENTS, MALWARE, VULNERABILITIES / TAGGED: "DEFAULTJSCRIPT", MALWARE

CYBER SECURITY CHANNEL

US GOVT WANTS NEW LABEL ON SECURE IOT DEVICES OR WANTS TO DISCOURAGE USE OF CHINESE IOT GADGETS

24,649,096,027 (24.65 BILLION) ACCOUNT USERNAMES AND PASSWORDS HAVE BEEN LEAKED BY CYBER CRIMINALS TILL

HOW CHINESE APT HACKERS STOLE LOCKHEED MARTIN F-35 FIGHTER PLANE TO DEVELOP ITS OWN J-20 STEALTH

NOW IN 2022

FIGHTER AIRCRAFT [VIDEO]

info@securitynewspaper.com Privacy Policy