MSHTML [Let's defend – Write-up]

2021's 0-Day MSHTML

file location: /root/Desktop/ChallengeFiles/Employee_W2_Form.docx
file location2: /root/Desktop/ChallengeFiles/Employees_Contact_Audit_Oct_2021.docx
file location3: /root/Desktop/ChallengeFiles/Work_From_Home_Survey.doc
file location4: /root/Desktop/ChallengeFiles/income_tax_and_benefit_return_2021.docx

| Start Investigation

```
root@ip-172-31-3-35:~/Desktop/Tools# python3 zipdump.py /root/Desktop/ChallengeFiles/Employees_Contact_Audit_Oct_2021.docx
Index Filename                      Encrypted Timestamp
    1 [Content_Types].xml                 0 1980-01-01 08:00:00
    2 _rels/                              0 2021-09-01 23:23:30
    3 _rels/.rels                         0 1980-01-01 08:00:00
    4 docProps/                           0 2021-09-01 23:23:30
    5 docProps/app.xml                    0 1980-01-01 08:00:00
    6 docProps/core.xml                   0 1980-01-01 08:00:00
    7 word/                               0 2021-09-01 23:23:30
    8 word/_rels/                         0 2021-09-01 23:23:30
    9 word/_rels/document.xml.rels        0 2021-09-10 16:37:10
   10 word/document.xml                   0 2021-09-01 23:23:30
   11 word/fontTable.xml                  0 1980-01-01 08:00:00
   12 word/media/                         0 2021-09-01 23:23:30
   13 word/media/image1.wmf               0 1980-01-01 08:00:00
   14 word/settings.xml                   0 1980-01-01 08:00:00
   15 word/styles.xml                     0 1980-01-01 08:00:00
   16 word/theme/                         0 2021-09-01 23:23:30
   17 word/theme/theme1.xml               0 1980-01-01 08:00:00
   18 word/webSettings.xml                0 1980-01-01 08:00:00
root@ip-172-31-3-35:~/Desktop/Tools#
```

To enhance our analysis, we can pipe the output into another powerful tool developed by Didier Stevens: re-search.py.

re-search.py is a specialized utility that leverages regular expressions (regex) to parse and search through data. It allows you to utilize a built-in library of common patterns or define your own custom expressions for more targeted searches.

By combining zipdump and re-search, we can execute a command that dumps all file indexes from the sample and pipes them directly into the search tool. We will then apply specific filters to automatically identify and extract all unique IPv4 addresses from the output.

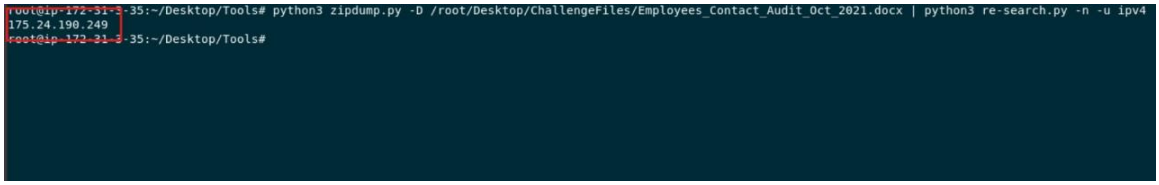
Q1: Examing the Employees_Contact_Audit_Oct_2021.docx file, what is the malicious IP in the docx file?

Analyzing OOXML Files with zipdump.py

Since .docx files are essentially compressed ZIP archives, we can explore their internal structure using zipdump.py (by Didier Stevens), as recommended in the SANS cheat sheet.

To examine the contents of the document, run the following command:

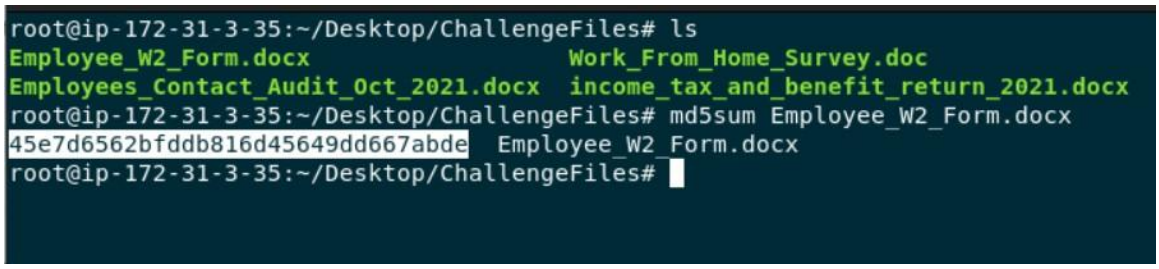
python3 zipdump.py /root/Desktop/ChallengeFiles/Employees_Contact_Audit_Oct_2021.docx

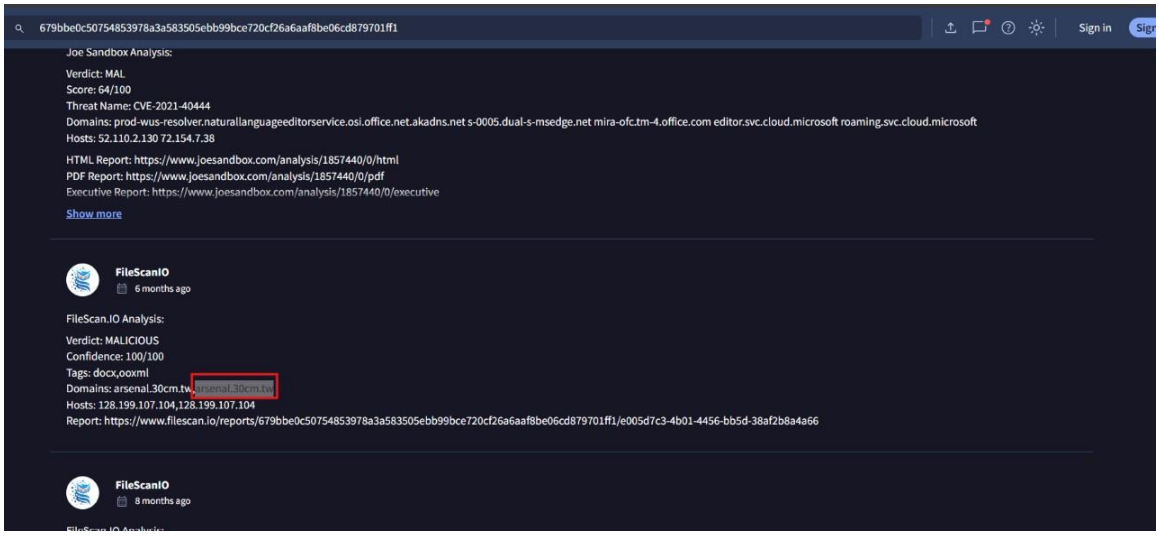


Answer: 175.24.190.249.

Q2: Examing the Employee_W2_Form.docx file, what is the malicious domain in the docx file?

To answer the question, I extracted the file hash and searched in Virustotal.

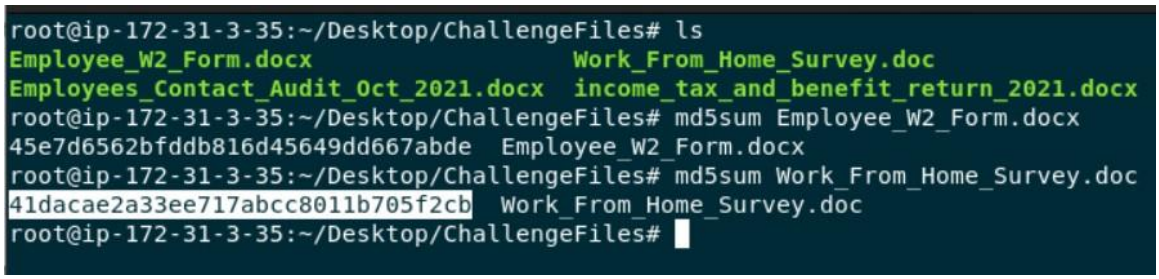


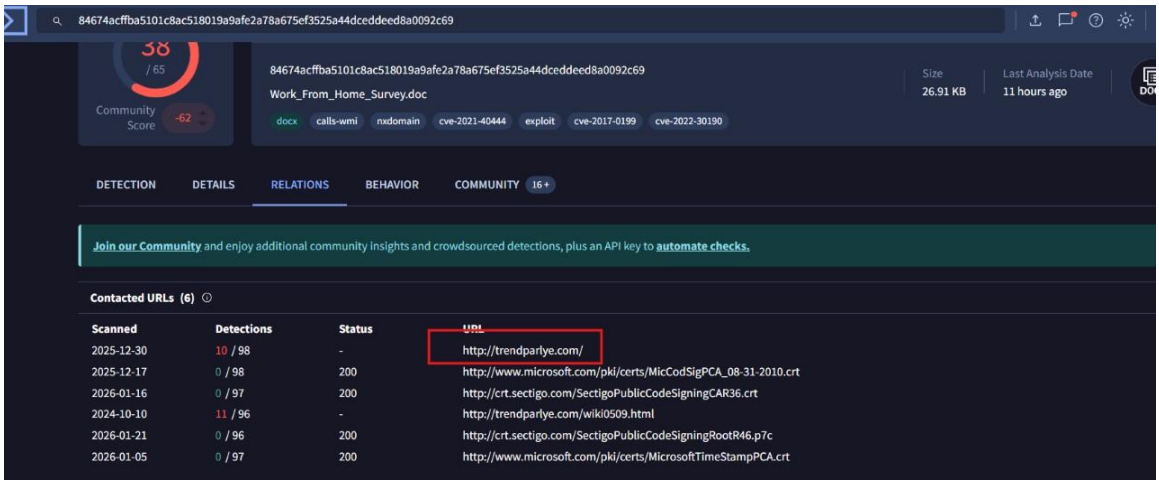


Answer: arsenal.30cm.tw

Q3: Examing the Work_From_Home_Survey.doc file, what is the malicious domain in the doc file?

Also to answer this question, I extracted the file hash and searched in Virustotal.

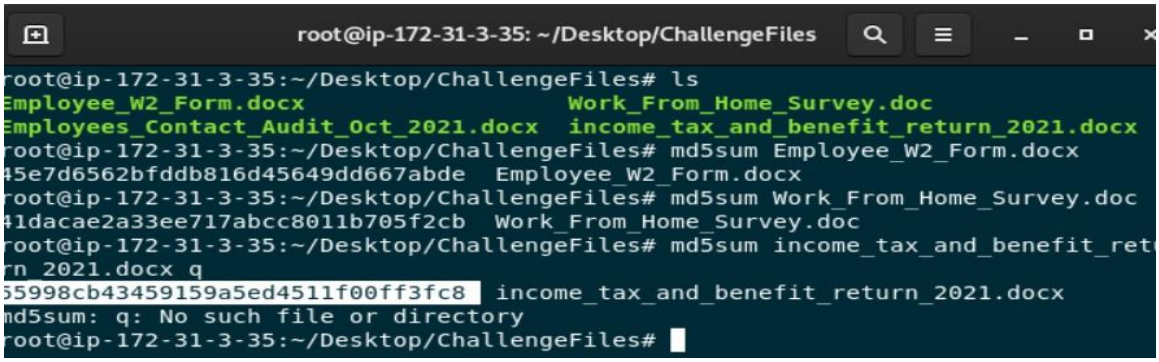


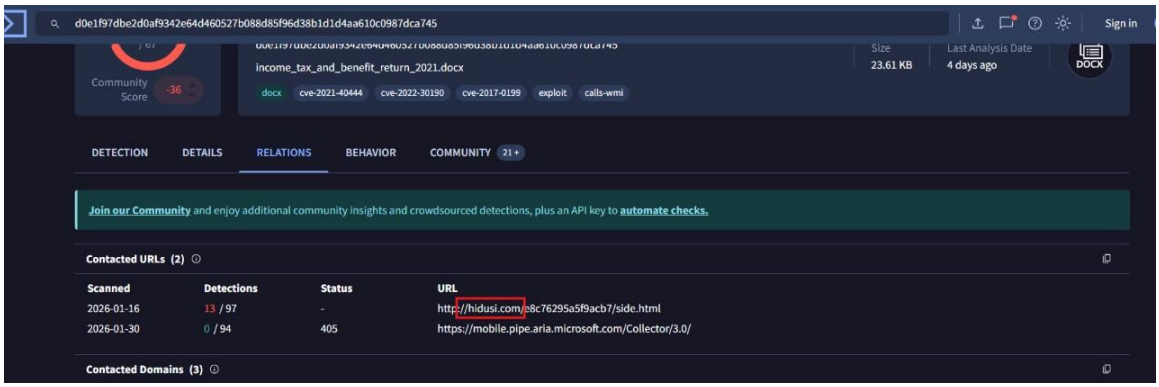


Answer: trendparlye.com

Q4: Examing the income_tax_and_benefit_return_2021.docx, what is the malicious domain in the docx file?

Also to answer this question, I extracted the file hash and searched in Virustotal.

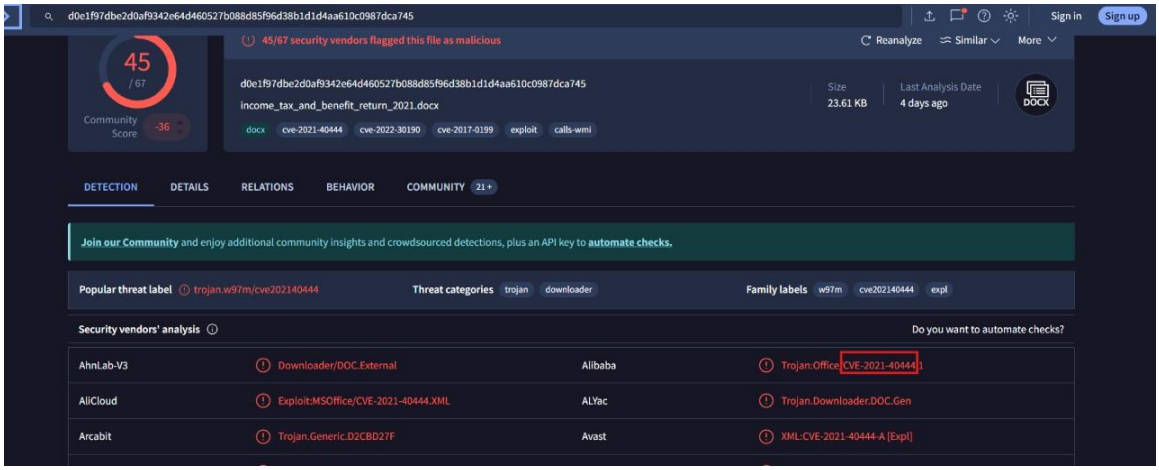




Answer: hidusi.com

Q5: What is the vulnerability the above files exploited?

The answer is also found in VirusTotal.



Answer: cve-2021-40444

Summary

By analyzing multiple samples like Employee_W2_Form.docx and Work_From_Home_Survey.doc, I learned how to use zipdump.py and re-search.py to peel back the layers of OOXML structures and extract hidden malicious IPs and domains. This lab was particularly beneficial because it demonstrated the "real-world" transition from a simple document to a network-based attack, showing me how to pivot from local file analysis to global threat intelligence tools like VirusTotal to identify specific vulnerabilities.