MuddyWater APT Group and A Methodology Proposal for Macro Malware Analysis

Araştırma Makalesi/Research Article

Mevlut Serkan TOK¹, Baris CELIKTAS²*

¹Computer Engineering Department, Graduate School of Engineering and Science, TOBB ETU, Ankara, Turkey
²Cyber Security Engineering and Cryptography, Institute of Informatics, ITU, Istanbul, Turkey

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 gruplarınınca da makro zararlı yazılımın atak vektörlerinde kullanılıldığı bilinmektedir. 2017 yılından beri Ortadoğu ülkelerinin kamu kurumları ve enerji, telekomünikasyon, petrol gibi stratejik alanlardan faaliyet gösteren şirketleri hedef alan, analistler tarafından kendilerini gizleme eğilimleri nedeniyle MuddyWater olarak adlandırılan ve Iran ile ilişkilendirilen ve Türkiye de dahil olmak üzere bölge ülkelerinde eylemlerini sürdürümektedir. Bu çalışmanın temel amacı MuddyWater grubunun özelliklerini, 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.

![Decoy Documents](image1.png)

**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].

![Spear Phishing Email](image2.png)

**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].

![Communication Infrastructure](image3.png)

**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>
<thead>
<tr>
<th>Date</th>
<th>Name</th>
<th>MD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.01.2018</td>
<td>2015 Yılı Ar-Ge Faaliyetleri Anketi Sonuçları.doc</td>
<td>781bd6b421a473206fc3 7919f28a27db</td>
</tr>
<tr>
<td>18.01.2018</td>
<td>ngn.tr.doc</td>
<td>faa44669d5cd90623312c 86d651fd3930</td>
</tr>
<tr>
<td>28.01.2018</td>
<td>KEGM-CyberAttack.doc</td>
<td>e87ea47e915407006310 825152dc802</td>
</tr>
<tr>
<td>28.02.2018</td>
<td>MIT.doc</td>
<td>fb8e6a0347a3af3dd2ab1b4e5a1be18a</td>
</tr>
</tbody>
</table>

### Table 1. The Group’s Malicious Documents
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 “form.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.

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 proexp.
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](image)

| 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 |

Table 2. The Metadata of the Malicious Document.

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](image)

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.

![Figure 8. The Macro Code Attached to Document](image)

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+hQSHTc7G2FgbWYoIvJv6M63uv 75asG5PirxxViQyYwzMr44VypWnDps64bU1SPyr8BG Jt7SOuWHt2bA7oEnE7kLgGdEwzZQkL9/” 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](image)

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 scrumpy expressions. Scrumpy and complete parameters are given below respectively.

```
pOwErShEll -NoEx -n0I -nOproFiLe -nOnIn -eXeCuTI BypAss -wiNdoWstYl h1DdEn -
```
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 hxsp://
microsofdatalinkpc.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.

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 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)

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 persistence mechanism in
terms of malware writing. Thus, this IOC was noted down
and “analyzing scheduled tasks” step was added to the
analysis plan.

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

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
grows 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.

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.

The script creates a file to path C:_USERS_\[username\]_APPDATA_LOCAL_TEMP named as rYFi1peADA.log and records activities and keys pressed (see Figure 17).

The script encodes log file and sends to “hxps://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.

The script creates a “rYFi1peADA” 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.

To provide persistence, “cscript” enables task scheduler COM API to create a scheduled task named as “rYFi1peADA” (see Figure 23). The details of the persistence mechanism are presented in section 3.8.

3.6.2. Stealing Cookies, Sessions, and Logins

The script 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.

Cscript creates a file to path C:_USERS_\[username\]_APPDATA_LOCAL_TEMP named as rYFi1peADA.log and records activities and keys pressed (see Figure 17).

The script encodes log file and sends to “hxps://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.

The script creates a “rYFi1peADA” 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.

To provide persistence, “cscript” enables task scheduler COM API to create a scheduled task named as “rYFi1peADA” (see Figure 23). The details of the persistence mechanism are presented in section 3.8.
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.

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

<table>
<thead>
<tr>
<th>File Path and Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Users[username]\AppDataLocal\Temp \lib_x64\System.Data.SQLite.dll</td>
<td>.dll</td>
</tr>
<tr>
<td>C:\Users[username]\AppDataLocal\Temp \lib_x64\System.Data.SQLite.Interop.dll</td>
<td>.dll</td>
</tr>
<tr>
<td>C:\Users[username]\AppDataLocal\Temp \201812041014 (Filename is created with timestamp of system)</td>
<td>sqlite</td>
</tr>
<tr>
<td>C:\Users[username]\AppDataLocal\Temp \yYF1pgeADA.xml</td>
<td>xml</td>
</tr>
<tr>
<td>C:\Users[username]\AppDataLocal\Temp \yYF1pgeADA.log</td>
<td>log</td>
</tr>
</tbody>
</table>

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 regsvr.32 (see Figure 26). This method is called spuiblydoo attack and was used in campaigns targeting governments before [28].

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.

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>
<thead>
<tr>
<th>File Name</th>
<th>MD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Festmara.doc (iistmar -&gt; Form)</td>
<td>bba017e5c34c1de3ef0b0d93195da70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MD5</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>3ab1d57658af32f23226001750d0231</td>
<td>hxpp://microsoftdata.linkpc.net/assesst/sqlite</td>
</tr>
<tr>
<td></td>
<td>hxpp://microsoftdata.linkpc.net/api/cscript</td>
</tr>
<tr>
<td></td>
<td>hxpp://microsoftdata.linkpc.net/api/pscript</td>
</tr>
<tr>
<td></td>
<td>hxpp://microsoftdata.linkpc.net/api/logger/submit</td>
</tr>
<tr>
<td></td>
<td>hxpp://microsoftdata.linkpc.net/api/opera/submit</td>
</tr>
<tr>
<td></td>
<td>hxpp://microsoftdata.linkpc.net/api/chrome/submit</td>
</tr>
<tr>
<td></td>
<td>hxpp://microsoftdata.linkpc.net/api/firefox/submit</td>
</tr>
</tbody>
</table>

Table 4. Revealed IOCs

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 “festival.doc” file as malicious (see Figure 28). Similarly, they recognize the “cscript” file as clean (see Figure 29).

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 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


