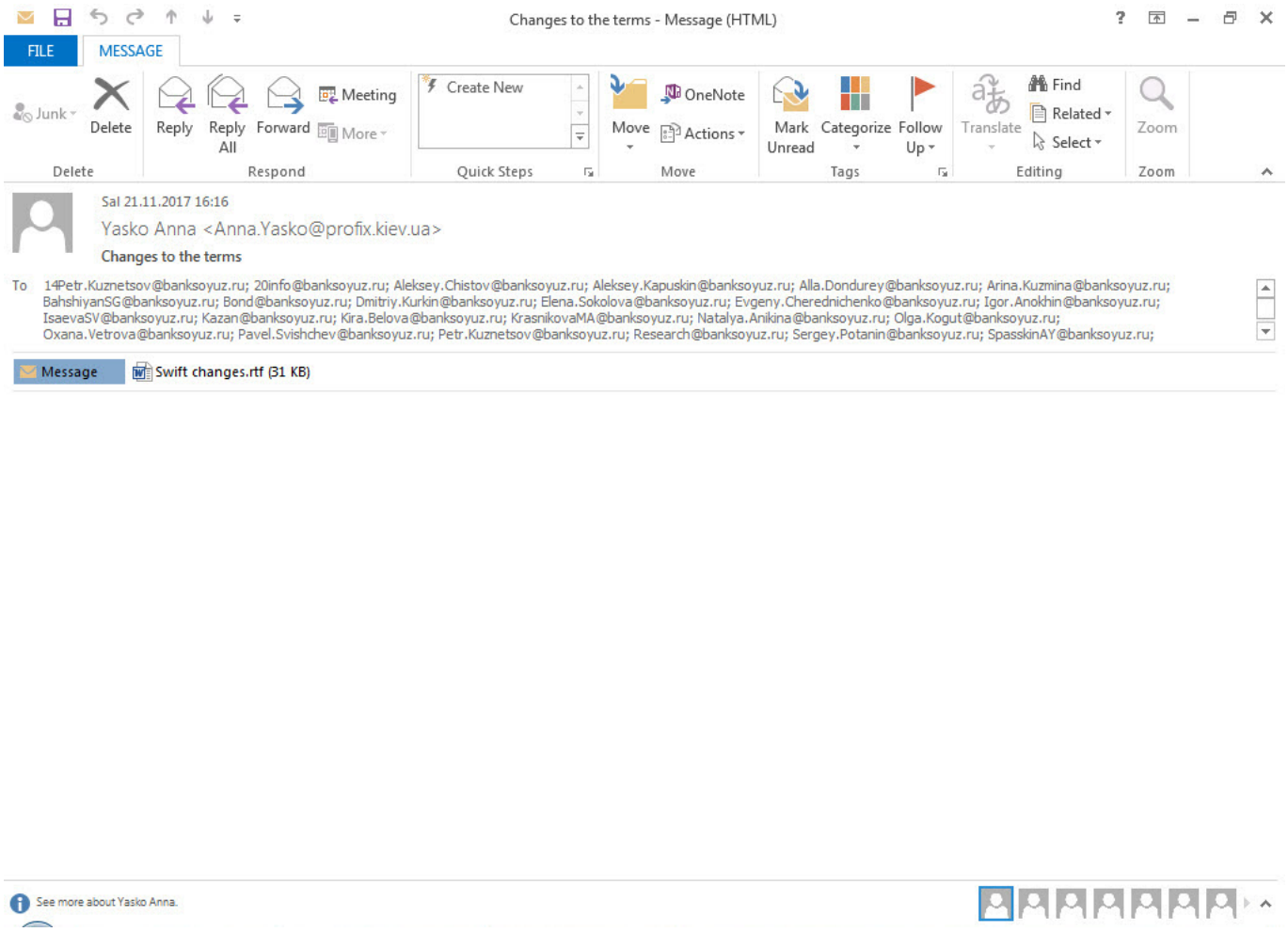
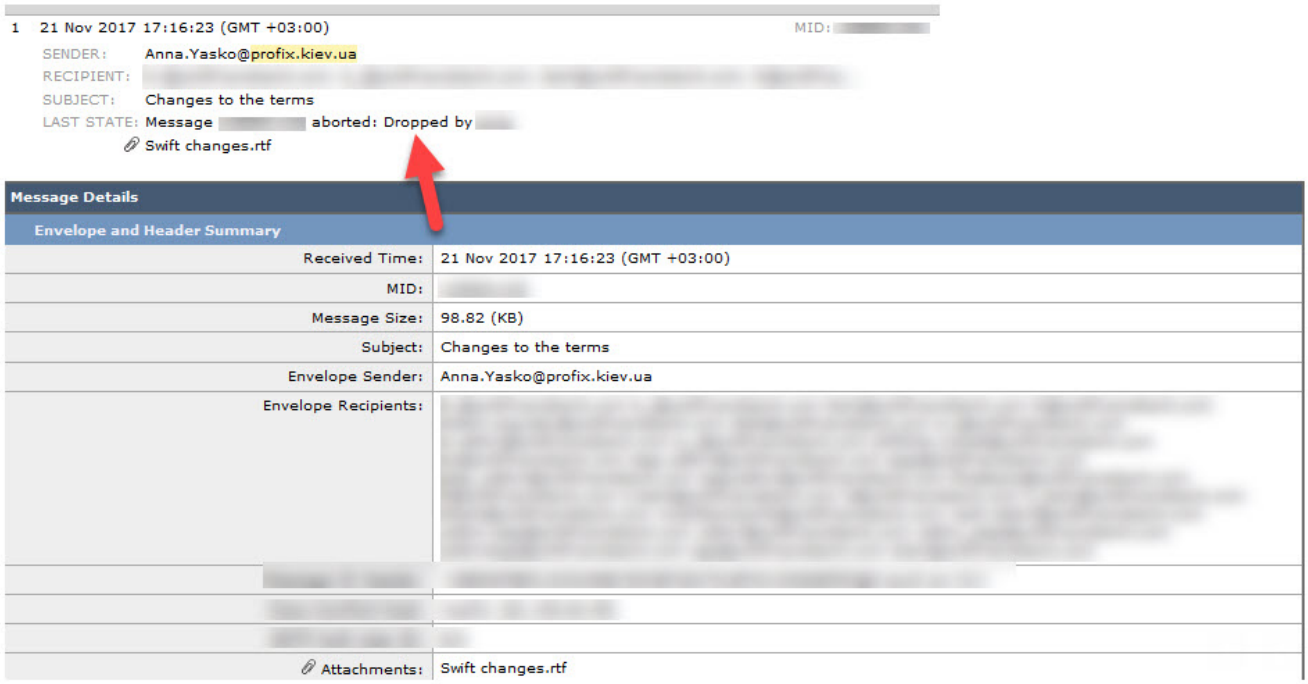
MetaStrike Operasyonu

written by Mert SARICA | 1 May 2018

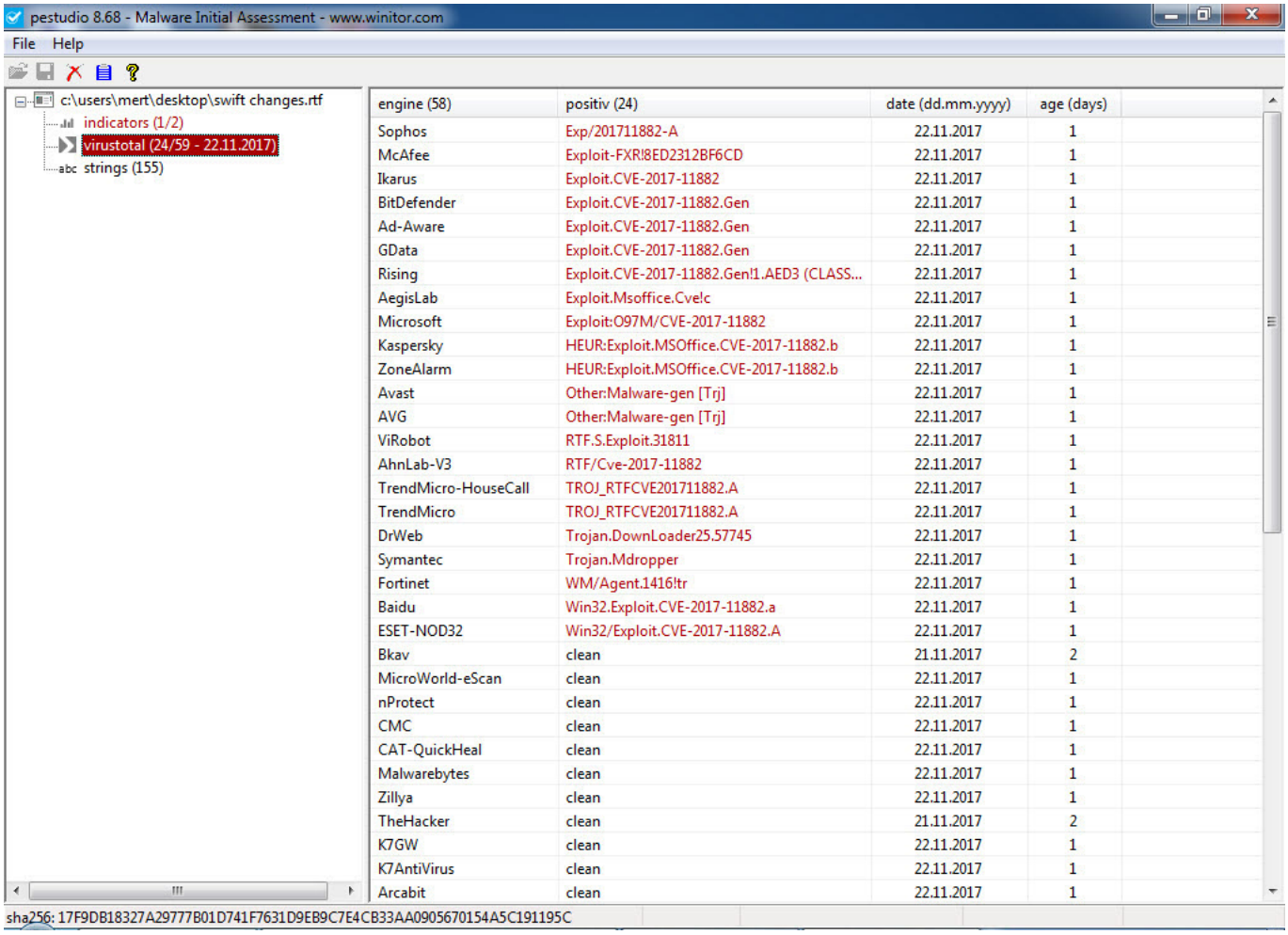
21.11.2017 tarihinde saat 16:16'da, Türkiye'de bulunan 19 bankanın toplam 535 çalışanına Anna.Yasko@profix.kiev.ua e-posta adresinden Changes to the terms başlığına sahip, ekinde Swift Changes.rtf dosyası bulunan bir e-posta gönderildi ve çoğu bankanın kullanmış olduğu güvenlik sistemleri tarafından ya silindi ya da karantinaya alındı.





Güvenlik sistemlerinde çok sayıda alarma yol açan bu şüpheli e-posta incelendiğinde, başlıkta ve ekindeki dosyada Swift kelimesinin geçmesinin yanısıra, alıcı listesinde (To:) yabancı banka çalışanlarının da dahil olduğu tam 952 kişinin yer alıyor olması ve e-postanın ileti gövdesinde (body) herhangi bir metnin yer almaması şüpheleri fazlasıyla arttırıyordu. E-postanın başlık bilgileri incelendiğinde, gönderen SMTP sunucusunun gerçekten de ProFIX firmasına ait olması ilk olarak bu kurumun hacklenmiş olabileceğine işaret ediyordu. Kim bu ProFIX diye ufak bir araştırma yapıldığında, 29 ülkede 250'den fazla banka ile çalışan, 2013 yılından bu yana ise Belarus, Ermenistan, Gürcistan, Ukrayna ve Moldovya'da hizmet veren bir SWIFT iş ortağı olduğu anlaşılıyordu.

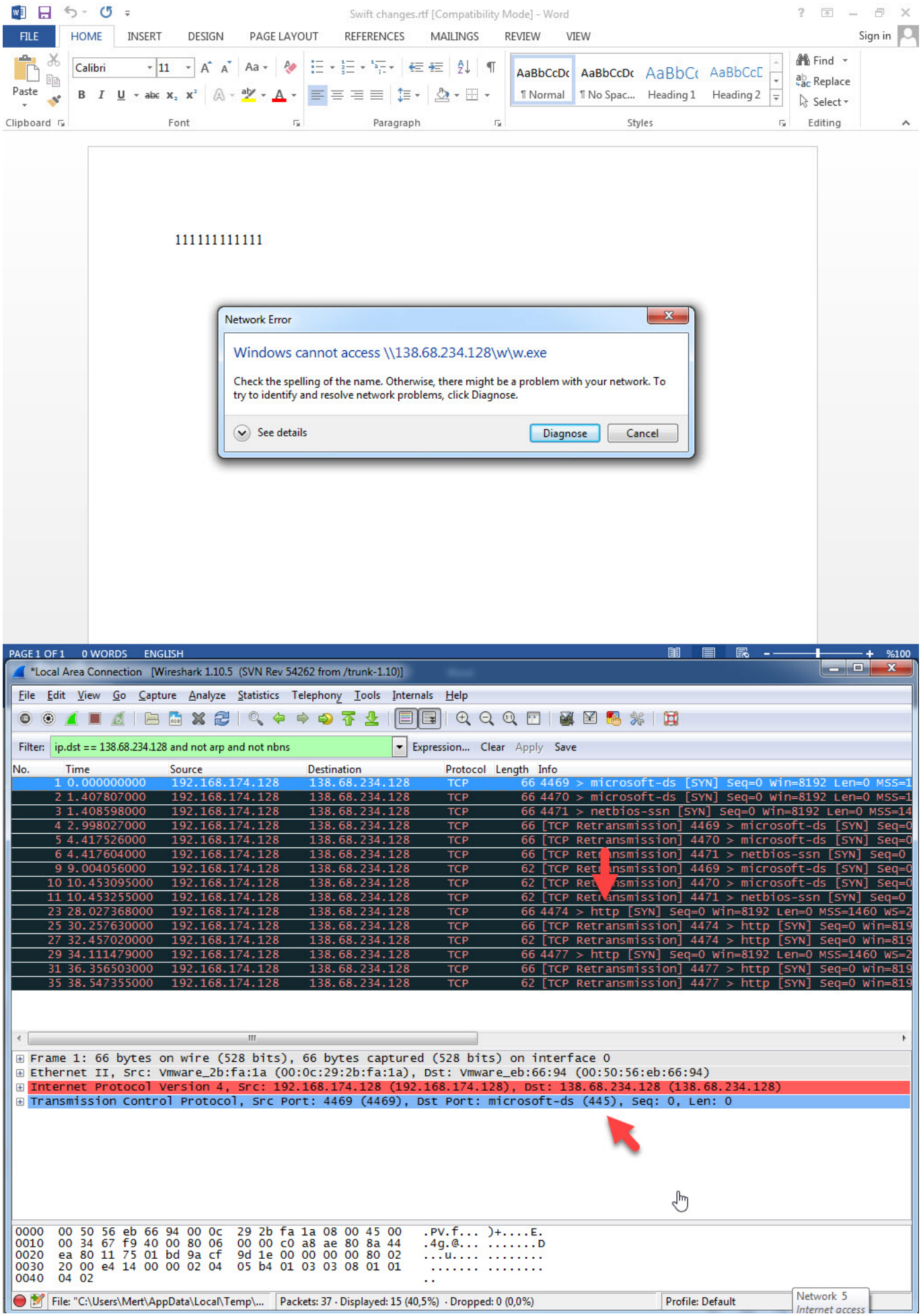
Swift Changes.rtf dosyası incelendiğinde ise bu dosyanın içinde Microsoft Office 2007'den 2016'ya kadar tüm sürümlerini etkileyen bir zafiyeti (CVE-2017-11882) istismar eden bir istismar kodu olduğu ortaya çıktı. 14 Kasım'da Microsoft tarafından yaması yayınlanan, GitHub üzerinde ise 20 Kasım'da istismar kodu yayınlanan bir zafiyet, 21 Kasım'da Türkiye'deki 19 bankaya siber saldırı gerçekleştirmek amacıyla, hacklendiği düşünülen ProFIX isimli bir SWIFT iş ortağı üzerinden gerçekleştiriliyordu!



Bulmacının kayıp parçalarını birleştirdikçe ortaya zamanlaması muazzam, senaryosu amatörce ((To: kısmında 952 kişinin olması, ileti gövdesinde metin olmaması vs.) kurgulanmış bir siber saldırı girişimi çıktı. 952 e-posta adresinin nereden ve nasıl temin edildiği sorusuna tam olarak yanıt bulunamasa da, sosyal medya üzerinde siber güvenlik uzmanlarından Huzeyfe ÖNAL ve Furkan ÇALIŞKAN'ın tespitlerine göre 25 Eylül tarihinde Pastebin sitesinde yer alan bir liste baz alınmıştı. 952 e-posta adresi ile Pastebin'de yer alan bu liste karşılaştırıldığında e-posta adreslerinin çok büyük bir oranının bu liste ile örtüşüyor olması, güvenlik uzmanlarını doğrular nitelikteydi.

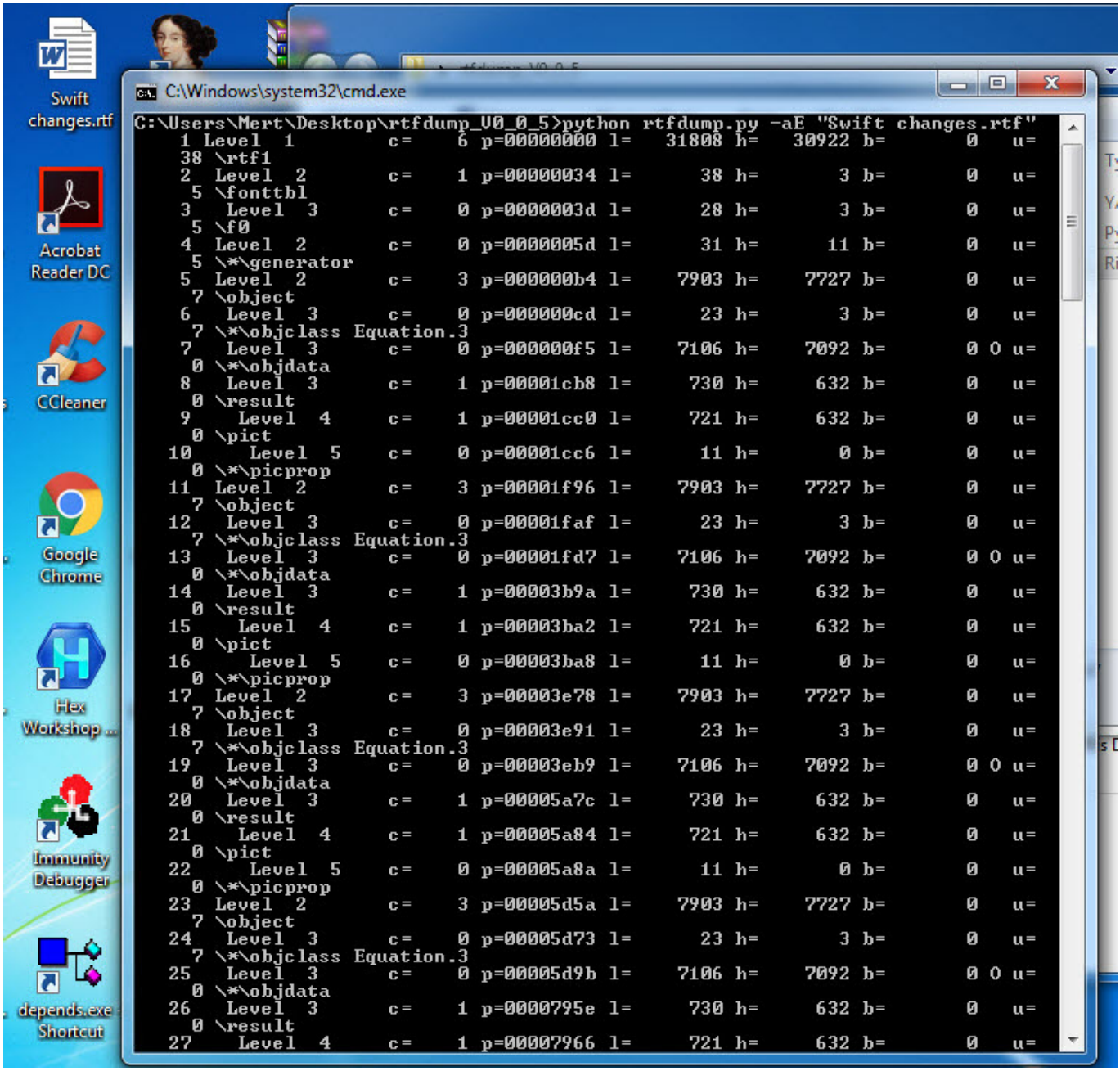
22 Kasım tarihinde Carbon Black firmasının blog sayfasında bu siber saldırının özet teknik detaylarına IOCleri ile birlikte yer verildi. 30 Kasım tarihli FireEye iSight istihbarat raporuna bakıldığında bu grubun 2016 yılından bu yana 19 ülkedeki finansal kurumları Cobalt Strike sızma testi yazılımı ile hedef alan Cobalt grubu (diğer bir adıyla MetaStrike) olduğu anlaşılıyordu. 8 Aralık tarihinde ise Palo Alto Networks firmasının blog sayfasında bu defa istismar kodunun teknik detaylarına yer verildi.

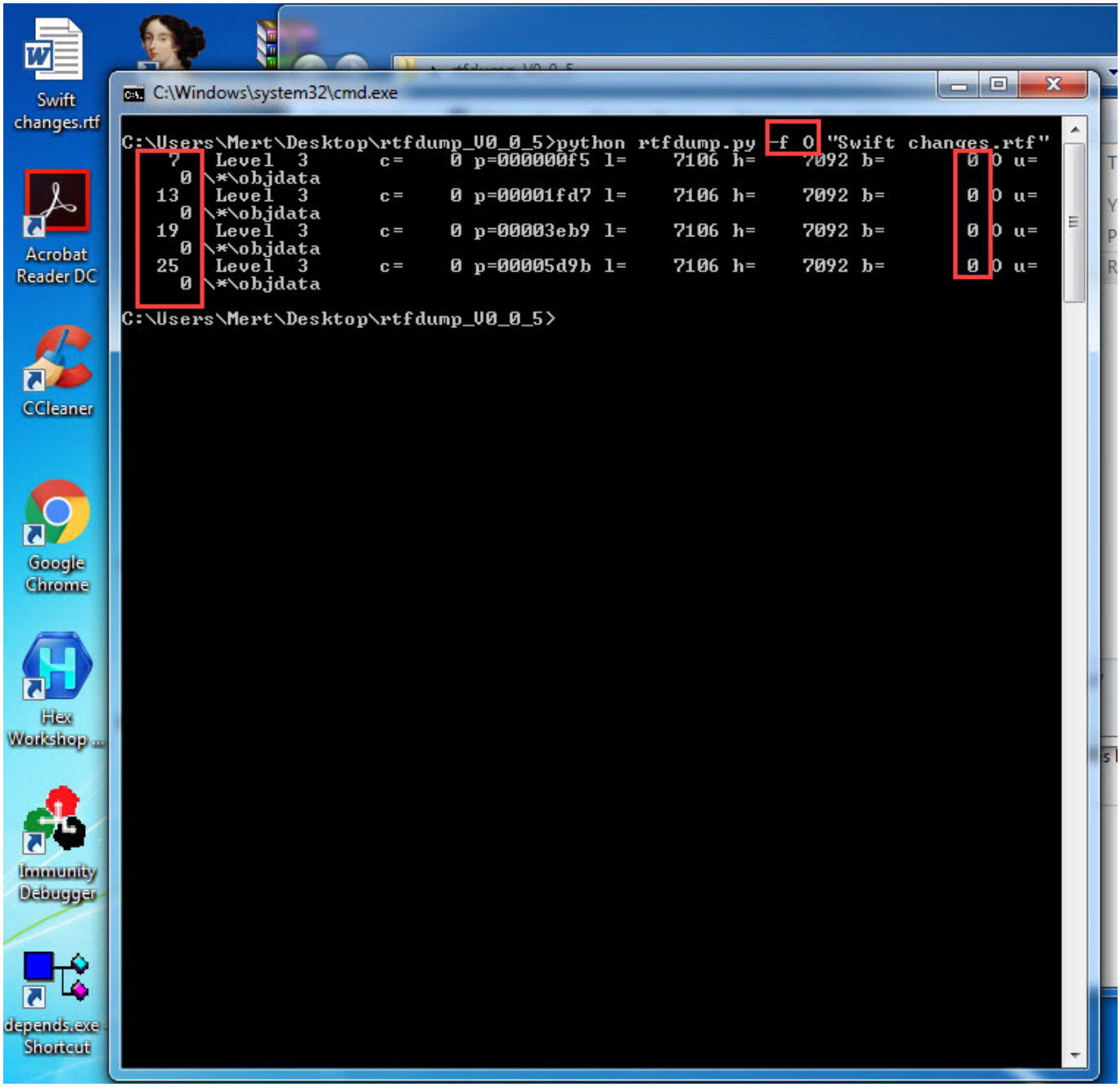
Swift Changes.rtf dosyası çalıştırılır çalıştırılmaz Microsoft Office'in Microsoft Equation Editor bileşenindeki yığın tabanlı bellek taşması (stack buffer overflow) zafiyetini istismar ederek \\138.68.234.12\w\w.exe paylaşım adresi üzerinden w.exe dosyasını çalıştırıyordu. Şayet Windows, SMB protokolü üzerinden ilgili adrese bağlanamıyor ise ve işletim sistemi üzerinde WebClient servisi çalışır durumda ise bu durumda WebDAV protokolü üzerinden tanımlı vekil sunucu (proxy) ayarlarını da dikkate alarak bağlanmaya çalışmaktaydı. Bu durum da istismar edilen hedef sistemin ilgili adrese erişip zararlı kodu çalıştırma ihtimalini fazlasıyla arttırıyordu!

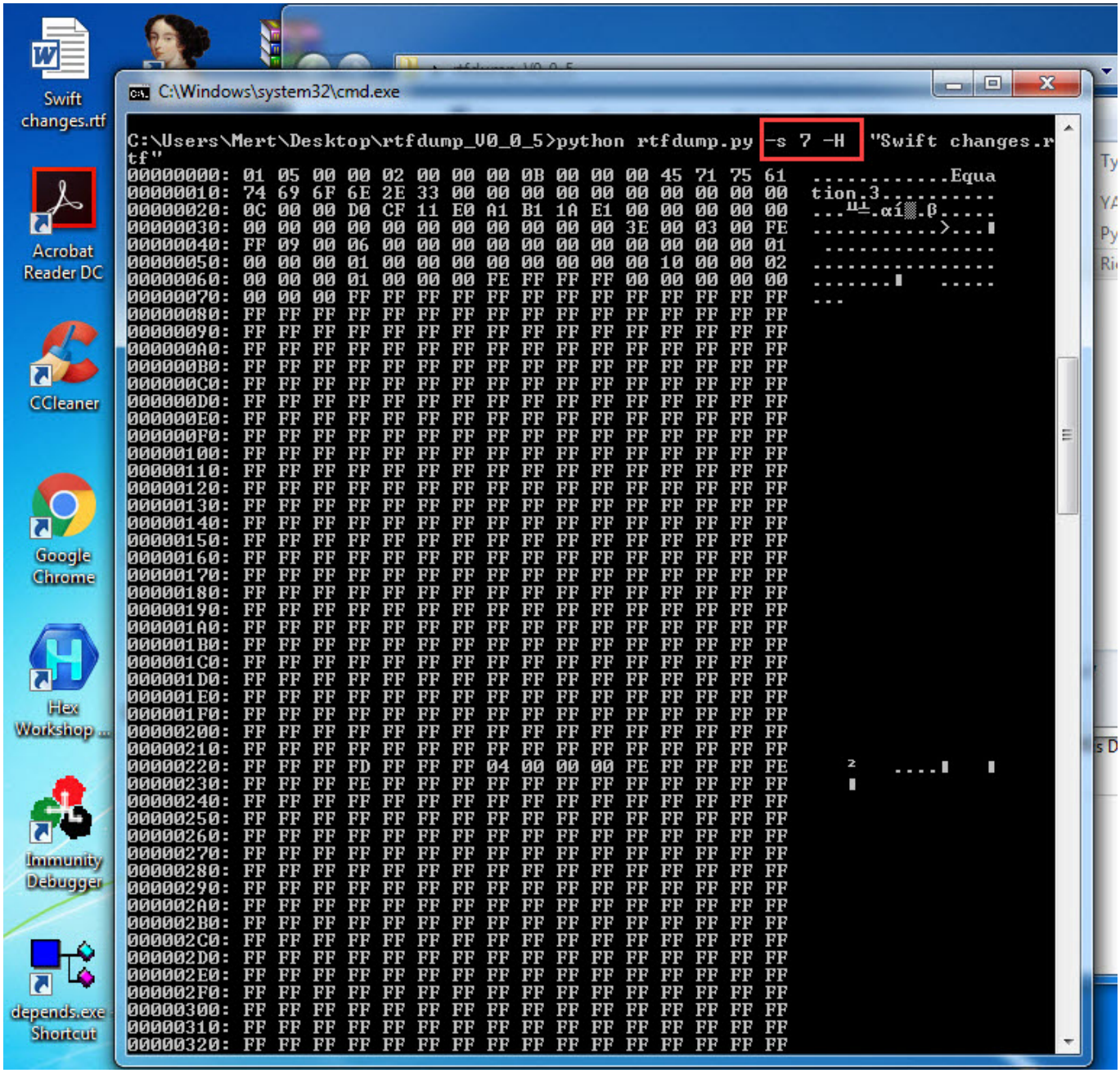

Swift changes.rtf [Compatibility Mode] - Word
FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW
Sign in
Paste
Calibri 11
Clipboard Font Paragraph Styles Editing
Normal No Spac... Heading 1 Heading 2
Find Replace Select
111111111111
Network Error
Windows cannot access \\138.68.234.128\w\w.exe
Check the spelling of the name. Otherwise, there might be a problem with your network. To try to identify and resolve network problems, click Diagnose.
See details
Diagnose
Cancel
PAGE 1 OF 1 0 WORDS ENGLISH %100
*Local Area Connection [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: ip.dst == 138.68.234.128 and not arp and not nbns
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000000 192.168.174.128 138.68.234.128 TCP 66 4469 > microsoft-ds [SYN] Seq=0 Win=8192 Len=0 MSS=1
2 1.407807000 192.168.174.128 138.68.234.128 TCP 66 4470 > microsoft-ds [SYN] Seq=0 Win=8192 Len=0 MSS=1
3 1.408598000 192.168.174.128 138.68.234.128 TCP 66 4471 > netbios-ssn [SYN] Seq=0 Win=8192 Len=0 MSS=14
4 2.998027000 192.168.174.128 138.68.234.128 TCP 66 [TCP Retransmission] 4469 > microsoft-ds [SYN] Seq=0
5 4.417526000 192.168.174.128 138.68.234.128 TCP 66 [TCP Retransmission] 4470 > microsoft-ds [SYN] Seq=0
6 4.417604000 192.168.174.128 138.68.234.128 TCP 66 [TCP Retransmission] 4471 > netbios-ssn [SYN] Seq=0
9 9.004056000 192.168.174.128 138.68.234.128 TCP 62 [TCP Retransmission] 4469 > microsoft-ds [SYN] Seq=0
10 10.453095000 192.168.174.128 138.68.234.128 TCP 62 [TCP Retransmission] 4470 > microsoft-ds [SYN] Seq=0
11 10.453255000 192.168.174.128 138.68.234.128 TCP 62 [TCP Retransmission] 4471 > netbios-ssn [SYN] Seq=0
23 28.027368000 192.168.174.128 138.68.234.128 TCP 66 4474 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
25 30.257630000 192.168.174.128 138.68.234.128 TCP 66 [TCP Retransmission] 4474 > http [SYN] Seq=0 Win=819
27 32.457020000 192.168.174.128 138.68.234.128 TCP 62 [TCP Retransmission] 4474 > http [SYN] Seq=0 Win=819
29 34.111479000 192.168.174.128 138.68.234.128 TCP 66 4477 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
31 36.356503000 192.168.174.128 138.68.234.128 TCP 66 [TCP Retransmission] 4477 > http [SYN] Seq=0 Win=819
35 38.547355000 192.168.174.128 138.68.234.128 TCP 62 [TCP Retransmission] 4477 > http [SYN] Seq=0 Win=819
Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Vmware_2b:fa:1a (00:0c:29:2b:fa:1a), Dst: Vmware_eb:66:94 (00:50:56:eb:66:94)
Internet Protocol Version 4, Src: 192.168.174.128 (192.168.174.128), Dst: 138.68.234.128 (138.68.234.128)
Transmission Control Protocol, Src Port: 4469 (4469), Dst Port: microsoft-ds (445), Seq: 0, Len: 0
0000 00 50 56 eb 66 94 00 0c 29 2b fa 1a 08 00 45 00 .PV.f...)+....E.
0010 00 34 67 f9 40 00 80 06 00 00 c0 a8 ae 80 8a 44 .4g.@... ....D
0020 ea 80 11 75 01 bd 9a cf 9d 1e 00 00 00 00 80 02 ...u.... ....
0030 20 00 e4 14 00 00 02 04 05 b4 01 03 03 08 01 01 ....
0040 04 02 ..
File: "C:\Users\Mert\AppData\Local\Temp\... Packets: 37 · Displayed: 15 (40,5%) · Dropped: 0 (0,0%) Profile: Default
Network 5
Internet access

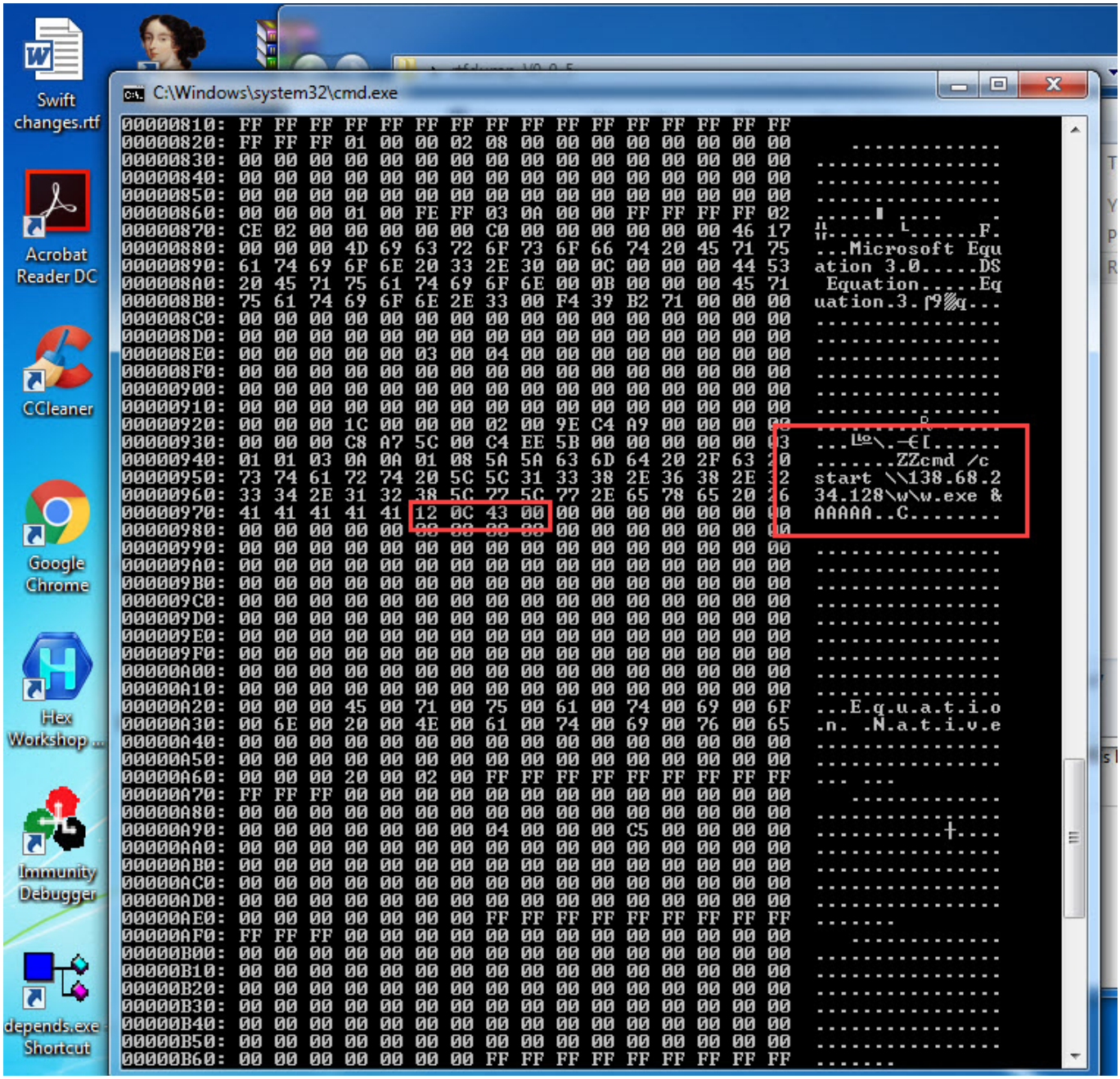
Son zamanlarda gerçekleştirilen siber saldırılarda zararlı RTF dosyalarının sıklıkla kullanılıyor olması sebebiyle bu yazı ile şüphe duyulan bir RTF dosyasının hızlı bir şekilde nasıl analiz edilebileceğine, Swift Changes.rtf dosyası özelinde yer vermek istedim. Zararlı RTF dosyalarının kötü emellerini gerçekleştirebilmeleri için OLE nesnelerinden faydalandıklarını bildiğimiz için Didier Stevens tarafından geliştirilen RTFDump aracı ile kısa bir sürede zararlı koda ulaşmak mümkün olabiliyor.

rtfdump aracına -aE parametresi vererek RTF içeriğini ASCII olarak görüntülediğimde ilgili OLE nesnelerini bulmak samanlıkla iğne aramaktan farksız olduğu için -f O parametresi ile sadece OLE nesnelerini listeledim. Ardından 7, 13, 19 ve 25 dizilerine tek tek -s ve -H (hex çıktısı) parametreleri ile baktığımda 7. dizide istismar kodu içine gömülü olan ip adresine ve Palo Alto Networks'un yazısına da konu olan WinExec fonksiyonunun adresine (0x430c12) statik olarak ulaştım.

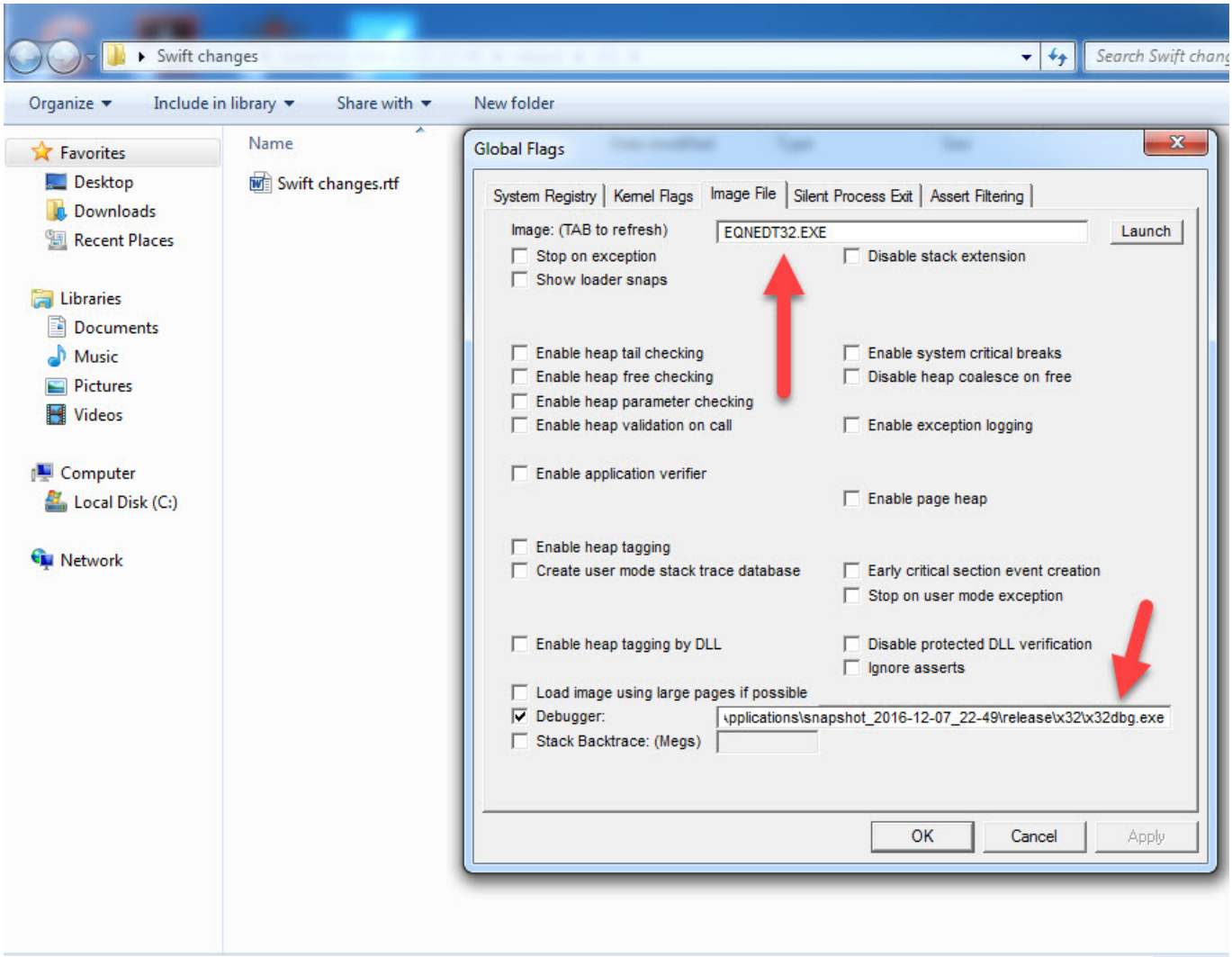

Swift changes.rtf
Acrobat Reader DC
CCleaner
Google Chrome
Hex Workshop ...
Immunity Debugger
depends.exe Shortcut
C:\Windows\system32\cmd.exe
C:\Users\Mert\Desktop\rtfdump_V0_0_5>python rtfdump.py -aE "Swift changes.rtf"
1 Level 1 c= 6 p=00000000 l= 31808 h= 30922 b= 0 u=
38 \rtf1
2 Level 2 c= 1 p=00000034 l= 38 h= 3 b= 0 u=
5 \fonttbl
3 Level 3 c= 0 p=0000003d l= 28 h= 3 b= 0 u=
5 \f0
4 Level 2 c= 0 p=0000005d l= 31 h= 11 b= 0 u=
5 *\generator
5 Level 2 c= 3 p=000000b4 l= 7903 h= 7727 b= 0 u=
7 \object
6 Level 3 c= 0 p=000000cd l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
7 Level 3 c= 0 p=000000f5 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
8 Level 3 c= 1 p=00001cb8 l= 730 h= 632 b= 0 u=
0 \result
9 Level 4 c= 1 p=00001cc0 l= 721 h= 632 b= 0 u=
0 \pict
10 Level 5 c= 0 p=00001cc6 l= 11 h= 0 b= 0 u=
0 *\picprop
11 Level 2 c= 3 p=00001f96 l= 7903 h= 7727 b= 0 u=
7 \object
12 Level 3 c= 0 p=00001faf l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
13 Level 3 c= 0 p=00001fd7 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
14 Level 3 c= 1 p=00003b9a l= 730 h= 632 b= 0 u=
0 \result
15 Level 4 c= 1 p=00003ba2 l= 721 h= 632 b= 0 u=
0 \pict
16 Level 5 c= 0 p=00003ba8 l= 11 h= 0 b= 0 u=
0 *\picprop
17 Level 2 c= 3 p=00003e78 l= 7903 h= 7727 b= 0 u=
7 \object
18 Level 3 c= 0 p=00003e91 l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
19 Level 3 c= 0 p=00003eb9 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
20 Level 3 c= 1 p=00005a7c l= 730 h= 632 b= 0 u=
0 \result
21 Level 4 c= 1 p=00005a84 l= 721 h= 632 b= 0 u=
0 \pict
22 Level 5 c= 0 p=00005a8a l= 11 h= 0 b= 0 u=
0 *\picprop
23 Level 2 c= 3 p=00005d5a l= 7903 h= 7727 b= 0 u=
7 \object
24 Level 3 c= 0 p=00005d73 l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
25 Level 3 c= 0 p=00005d9b l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
26 Level 3 c= 1 p=0000795e l= 730 h= 632 b= 0 u=
0 \result
27 Level 4 c= 1 p=00007966 l= 721 h= 632 b= 0 u=


Swift changes.rtf
Acrobat Reader DC
CCleaner
Google Chrome
Hex Workshop ...
Immunity Debugger
depends.exe Shortcut
C:\Windows\system32\cmd.exe
C:\Users\Mert\Desktop\rtfdump_V0_0_5>python rtfdump.py -f 0 "Swift changes.rtf"
7 Level 3 c= 0 p=000000f5 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
13 Level 3 c= 0 p=00001fd7 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
19 Level 3 c= 0 p=00003eb9 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
25 Level 3 c= 0 p=00005d9b l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
C:\Users\Mert\Desktop\rtfdump_V0_0_5>

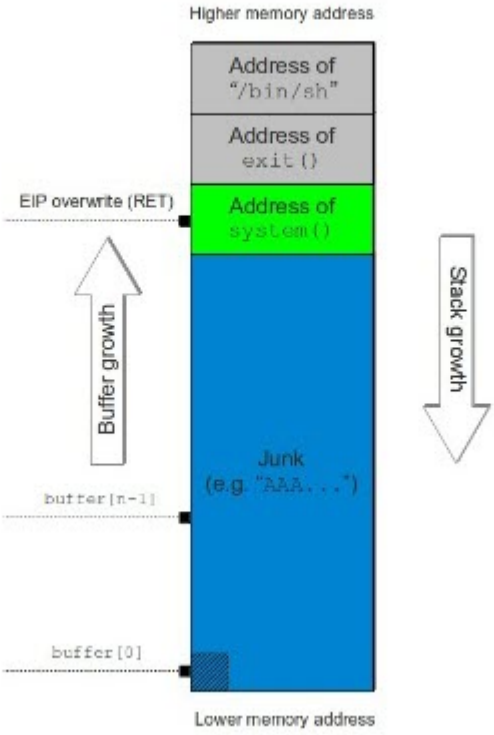

Swift changes.rtf
Acrobat Reader DC
CCleaner
Google Chrome
Hex Workshop ...
Immunity Debugger
depends.exe - Shortcut
C:\Windows\system32\cmd.exe
C:\Users\Mert\Desktop\rtfdump_V0_0_5>python rtfdump.py -s 7 -H "Swift changes.rtf"
00000000: 01 05 00 00 02 00 00 00 0B 00 00 00 45 71 75 61 ....Equa
00000010: 74 69 6F 6E 2E 33 00 00 00 00 00 00 00 00 00 00 tion.3..........
00000020: 0C 00 00 D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 ...Ð.à¡.á.á.....
00000030: 00 00 00 00 00 00 00 00 00 00 00 00 3E 00 03 00 ....>...
00000040: FE FF 09 00 06 00 00 00 00 00 00 00 00 00 00 00 ....
00000050: 01 00 00 00 01 00 00 00 00 00 00 00 10 00 00 00 ....
00000060: 02 00 00 00 01 00 00 00 FE FF FF FF 00 00 00 00 ....þÿÿÿ....
00000070: 00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF ...
00000080: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000090: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000100: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000110: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000120: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000130: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000140: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000150: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000160: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000170: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000180: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000190: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000001F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000200: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000210: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000220: FF FF FF FD FF FF FF 04 00 00 00 FE FF FF FF FE ....ý.....þÿÿÿ
00000230: FF FF FF FE FF FF FF FF FF FF FF FF FF FF FF FF þÿÿÿ
00000240: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000250: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000260: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000270: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000280: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000290: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000002A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000002B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000002C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000002D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000002E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000002F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000300: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000310: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000320: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

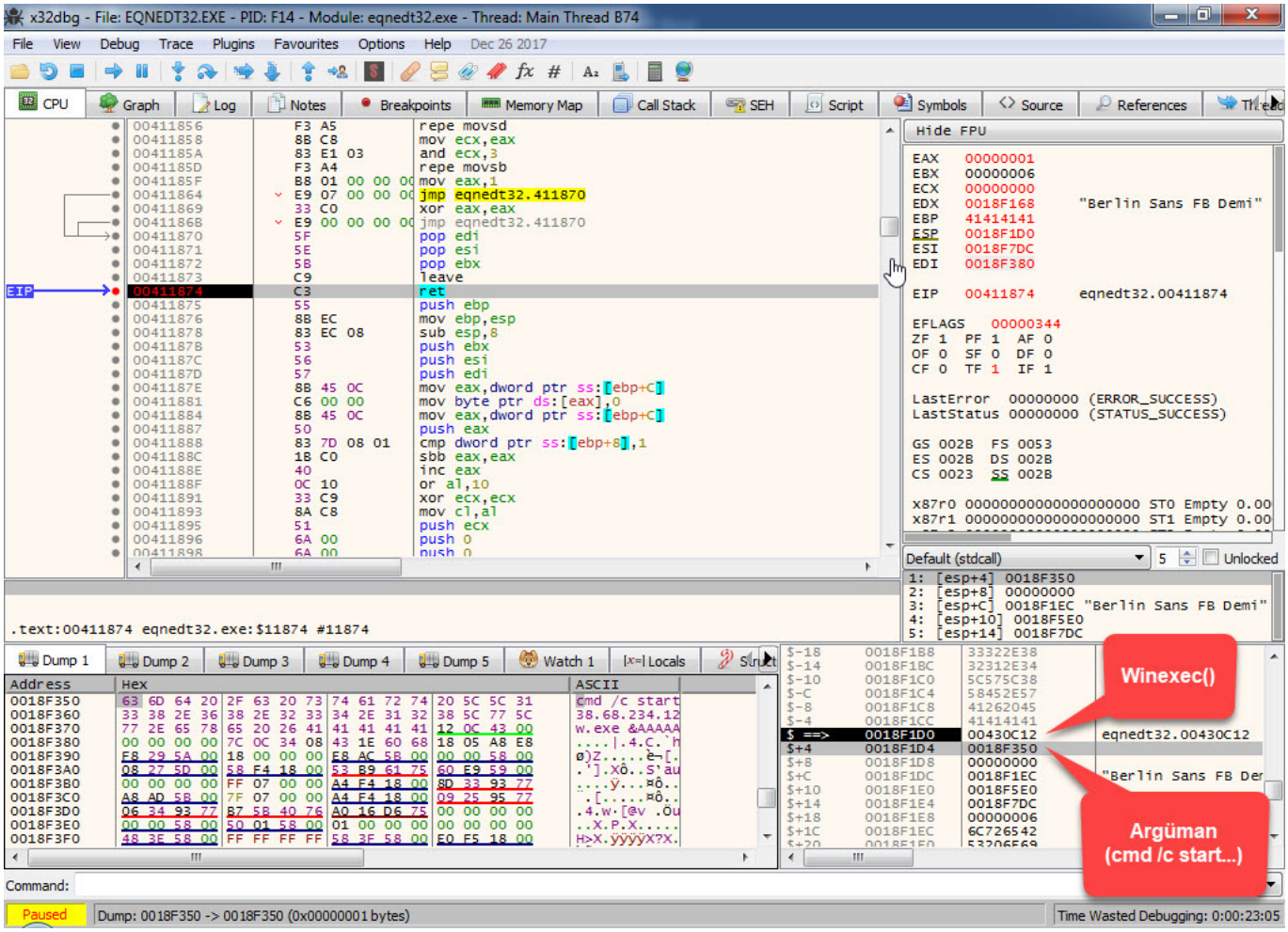


Dinamik kod analizi ile de doğrulamak için ise öncelikle Windows Debugging Tools ile gelen Global Flags Editor üzerinde bir ayarlama yapmam gerekti. Swift Changes.rtf dosyası Microsoft Office Word ile açıldıktan sonra Microsoft Equation Editor bileşenine ait eqnedt32.exe programını istismar ettiği için eqnedt32.exe programı açılır açılmaz x64dbg hata ayıklayıcının devreye gireceği şekilde ayarladım. Winexec fonksiyonun adresine kesme noktası koyup adım adım geriye gidip fonksiyon çıkışlarına da kesme noktası koyarak kısa sürede ret2libc yönteminden faydalanan istismar koduna ulaşmış oldum.



ret2libc (visual)





Sonuç olarak siber saldırı girişimi, hazırlanan e-postanın tahminimce aceleye gelmesi (To: kısmında 952 kişinin olması, ileti gövdesinde metin olmaması vs.) ve sıfırıncı gün zafiyeti yerine genele duyurulan bir istismar kodunun (bu sayede güvenlik üreticileri imzalarını hızlıca güncelleyebildiler.) kullanılması sayesinde bildiğim kadarıyla Türkiye'deki herhangi bir bankada başarıya ulaşamadı. İstismar kodunun genele açık olarak yayınlanmasından 1 gün sonra bankalarımıza gerçekleştirilen bu siber saldırı bankalarımızı, finans kurumlarımızı hedef alan grupların ne kadar hızlı bir şekilde organize olup, hareket etmesi gelecekte benzer girişimlere, dikkat edilmesi gereken hususlara dair önemli ipuçları veriyor. Yazıma son noktayı koymadan önce bu ve benzeri siber saldırı girişimlerinin başarıya ulaşmasını zorlaştırma adına finans kurumlarının, iç ağdan internete doğru WebDAV kullanımını engellemelerinde fayda olacağına inanıyorum.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: Kurum içinden internete doğru WebDAV bağlantısını kontrol etmek için Explorer üzerinden \\live.sysinternals.com adresine gitmeyi deneyebilirsiniz.