

# Expiro Malware Is Back and Even Harder to Remove

---

 [mcafee.com/blogs/other-blogs/mcafee-labs/expiro-infected-files-to-complicate-repair/](https://mcafee.com/blogs/other-blogs/mcafee-labs/expiro-infected-files-to-complicate-repair/)

October 31, 2017

[Xiaobing Lin](#)

Oct 31, 2017

5 MIN READ

File infector malware adds malicious code to current files. This makes removal tricky because deleting infections results in the loss of legitimate files. Although file infectors were more popular in the 1990s and early 2000s, they still pose a significant threat. The complex disinfection process is usually leveraged by malware authors to ensure systems stay infected for a long period. This may explain why complex file infectors such as W32/VirRansom, W32/Sality, W32/Xpaj, and Expiro are still active today.

The Expiro virus has been around for more than a decade, and the authors continue to update it with more features. Expiro is unique in that it infiltrates executable files on both 32- and 64-bit Windows systems by appending its viral code to the host. It can be used to install malicious browser extensions, lower browser security settings, and steal account credentials.

Recently we discovered a new variant of Expiro with a significant change in its infection routine. In previous variants, Expiro modified and stole code at the entry point and appended the viral payload only at the end of the original file, typical of an appender virus.

The new variant, however, changes the size of the base relocation table and encrypts the addresses inside, causing traditional appender virus repair routines to corrupt files unless they correctly restore the original base relocation table. By adding the encryption, Expiro increases the complexity of analysis and requires a customized repair routine, which makes it hard to combat.

The following screenshots demonstrate this point: The base relocation table of a file infected by the old variant of Expiro is unaffected and the contents are untouched.

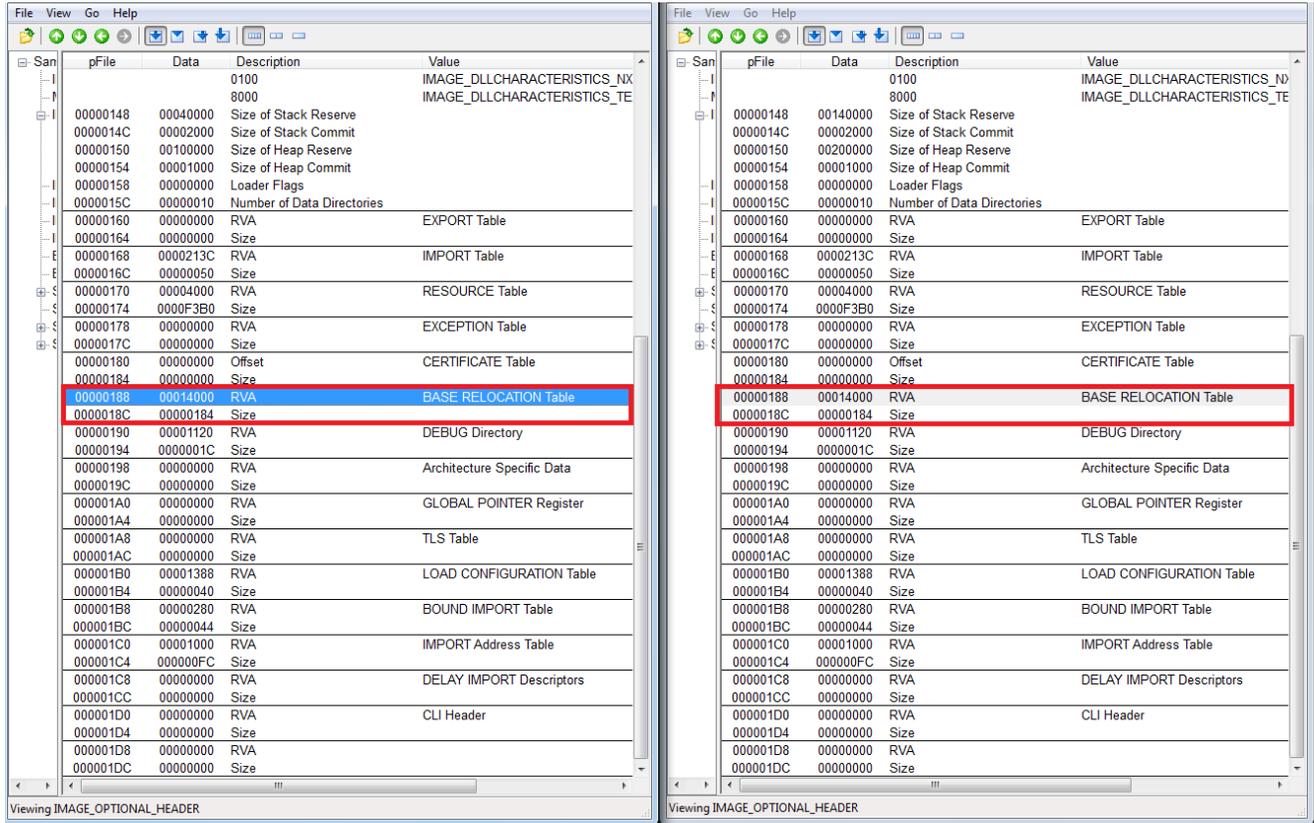


Figure 1: The relocation table remains intact when infected by the old Expiro variant.

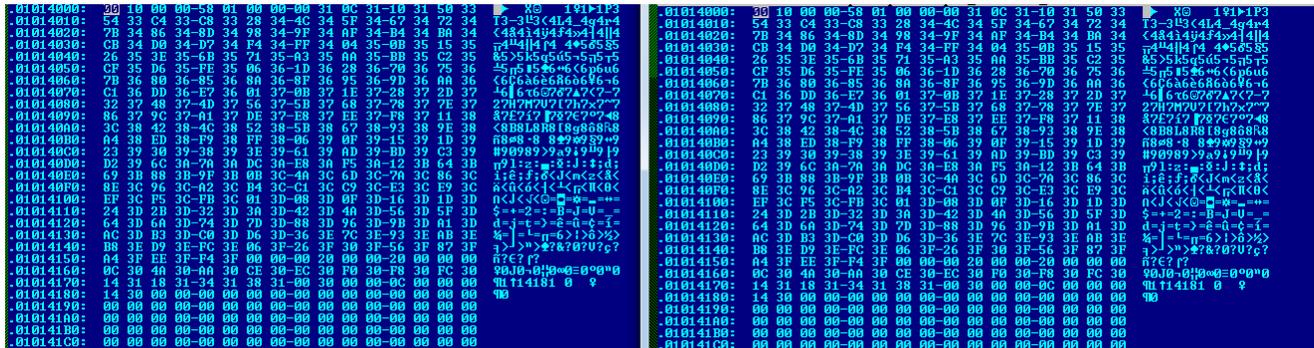


Figure 2: The relocation table contents are not modified by the old Expiro variant.

The new variant reduces the size of the base relocation table and encrypts portions of it (outlined in red).

pFile	Data	Description	Value
	0100	0100	IMAGE_DLLCHARACTERISTIC
	8000	8000	IMAGE_DLLCHARACTERISTIC
00000148	00040000	Size of Stack Reserve	
0000014C	00002000	Size of Stack Commit	
00000150	00100000	Size of Heap Reserve	
00000154	00001000	Size of Heap Commit	
00000158	00000000	Loader Flags	
0000015C	00000010	Number of Data Directories	
00000160	00000000	RVA	EXPORT Table
00000164	00000000	Size	
00000168	0000213C	RVA	IMPORT Table
0000016C	00000050	Size	
00000170	00004000	RVA	RESOURCE Table
00000174	0000F3B0	Size	
00000178	00000000	RVA	EXCEPTION Table
0000017C	00000000	Size	
00000180	00000000	Offset	CERTIFICATE Table
00000184	00000000	Size	
00000188	00014000	RVA	BASE RELOCATION Table
0000018C	00000184	Size	
00000190	00001120	RVA	DEBUG Directory
00000194	0000001C	Size	
00000198	00000000	RVA	Architecture Specific Data
0000019C	00000000	Size	
000001A0	00000000	RVA	GLOBAL POINTER Register
000001A4	00000000	Size	
000001A8	00000000	RVA	TLS Table
000001AC	00000000	Size	
000001B0	00001388	RVA	LOAD CONFIGURATION Table
000001B4	00000040	Size	
000001B8	00000280	RVA	BOUND IMPORT Table
000001BC	00000044	Size	
000001C0	00001000	RVA	IMPORT Address Table
000001C4	000000FC	Size	
000001C8	00000000	RVA	DELAY IMPORT Descriptors
000001CC	00000000	Size	
000001D0	00000000	RVA	CLI Header
000001D4	00000000	Size	
000001D8	00000000	RVA	
000001DC	00000000	Size	

Figure 3: The latest Expiro variant reduces the size of the relocation table.

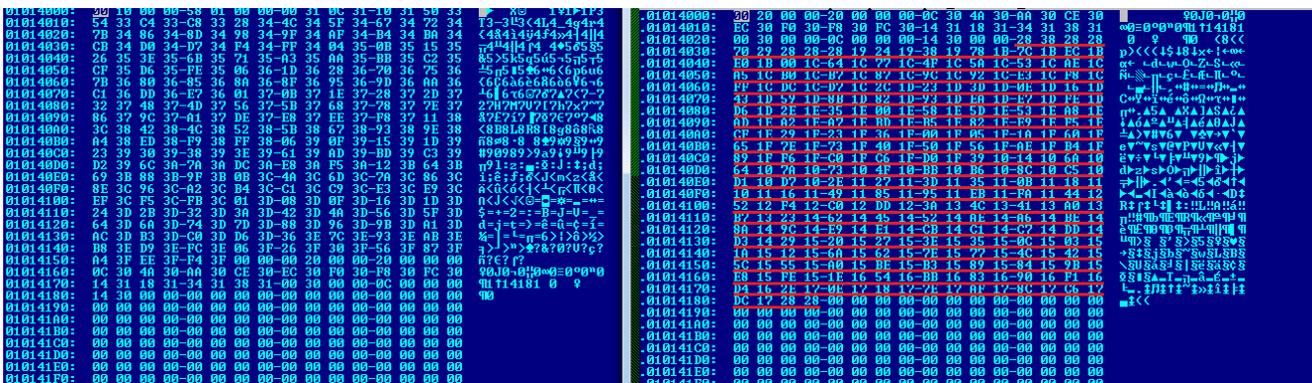


Figure 4: The relocation table encrypted by the latest Expiro variant.

To fix relocations prior to the execution of the original file's code, the Expiro virus first executes its own malicious payload. It then decrypts the relocation table and dynamically reloads all addresses to make sure the original file can run correctly.

Decryption involves a simple XOR operation with a key hardcoded within the sample.

A1 E8509B00	mov eax, dword ptr ds:[9B50E8]		Registers (MMX)
0305 9C519B00	add eax, dword ptr ds:[9B519C]		EAX 0096403C odbcad32.0096403C
83E8 04	sub eax, 4		ECX 00000000
0345 C4	add eax, dword ptr ss:[ebp-3C]		EDX 00953048 odbcad32.00953048
0345 C8	add eax, dword ptr ss:[ebp-38]		EBX 0097BD79 odbcad32.0097BD79
8985 74FFFFFF	mov dword ptr ss:[ebp-8C], eax		ESP 001BF4BC ASCII "cR"
8B7D EC	mov edi, dword ptr ss:[ebp-14]		EBP 001BF848
83EF 0B	sub edi, 0B		ESI 00002828
89C6	mov esi, eax		EDI 00001B7C — Encrypted data
0FB73C7E	movzx edi, word ptr ds:[esi+edi*2]	Read the Encrypted Relocation Table	EIP 009B0CEA odbcad32.009B0CEA
0FB775 96	movzx esi, word ptr ss:[ebp-6A]	Get Decryption Key "0x28"	C 0 ES 0023 32bit 0(FFFFFFFF)
31F7	xor edi, esi	XOR with key	P 1 CS 001B 32bit 0(FFFFFFFF)
66:897D AE	mov word ptr ss:[ebp-52], di		A 0 SS 0023 32bit 0(FFFFFFFF)
0FB77D AE	movzx edi, word ptr ss:[ebp-52]		Z 1 DS 0023 32bit 0(FFFFFFFF)
8B4D E0	mov ecx, dword ptr ss:[ebp-20]		S 0 FS 003B 32bit 7FFDF000(FFF)
83E9 08	sub ecx, 8		T 0 GS 0000 NULL
D3E7	shl edi, cl		D 0
66:897D BA	mov word ptr ss:[ebp-46], di		O 0 LastErr ERROR_SXS_KEY_NOT_FOUND (000036B7)
0FB77D BA	movzx edi, word ptr ss:[ebp-46]		

Figure 5: Relocation table being decrypted using a hardcoded XOR key.

After the decryption, the rest of original base relocation table is recovered.

A1 E8509B00	mov eax, dword ptr ds:[9B50E8]		Registers (MMX)
0305 9C519B00	add eax, dword ptr ds:[9B519C]		EAX 0096403C odbcad32.0096403C
83E8 04	sub eax, 4		ECX 00000000
0345 C4	add eax, dword ptr ss:[ebp-3C]		EDX 00953048 odbcad32.00953048
0345 C8	add eax, dword ptr ss:[ebp-38]		EBX 0097BD79 odbcad32.0097BD79
8985 74FFFFFF	mov dword ptr ss:[ebp-8C], eax		ESP 001BF4BC ASCII "cR"
8B7D EC	mov edi, dword ptr ss:[ebp-14]		EBP 001BF848
83EF 0B	sub edi, 0B		ESI 00002828
89C6	mov esi, eax		EDI 00000000 — Decrypted Data
0FB73C7E	movzx edi, word ptr ds:[esi+edi*2]	Read the Encrypted Relocation Table	EIP 009B0CEC odbcad32.009B0CEC
0FB775 96	movzx esi, word ptr ss:[ebp-6A]	Get Decryption Key "0x28"	C 0 ES 0023 32bit 0(FFFFFFFF)
31F7	xor edi, esi	XOR with key	P 0 CS 001B 32bit 0(FFFFFFFF)
66:897D AE	mov word ptr ss:[ebp-52], di		A 0 SS 0023 32bit 0(FFFFFFFF)
0FB77D AE	movzx edi, word ptr ss:[ebp-52]		Z 0 DS 0023 32bit 0(FFFFFFFF)
8B4D E0	mov ecx, dword ptr ss:[ebp-20]		S 0 FS 003B 32bit 7FFDF000(FFF)
83E9 08	sub ecx, 8		T 0 GS 0000 NULL
D3E7	shl edi, cl		D 0
66:897D BA	mov word ptr ss:[ebp-46], di		O 0 LastErr ERROR_SXS_KEY_NOT_FOUND (000036B7)
0FB77D BA	movzx edi, word ptr ss:[ebp-46]		

Figure 6: The EDI register now contains decrypted relocation data.

In recovery step 2, Expiro computes the address that contains the relocation address using the formula  $Relocation\_Address = NewImageBase + Offset + VirtualAddress$ .

A1 F4509B00	mov eax, dword ptr ds:[9B50F4]		Registers (MMX)
0305 F4519B00	add eax, dword ptr ds:[9B51F4]		EAX 00950354 odbcad32.00950354
83E8 06	sub eax, 6		ECX 00000000
0FB795 72FFFFFF	movzx edx, word ptr ss:[ebp-8E]		EDX 00000354
39D0	cmp eax, edx		EBX 0097BD79 odbcad32.0097BD79
75 32	jnz short 009B0DAC		ESP 001BF4BC ASCII "cR"
8B45 D0	mov eax, dword ptr ss:[ebp-30]	New_ImageBase	EBP 001BF848
0FB755 BA	movzx edx, word ptr ss:[ebp-46]	Offset	ESI 0000000B
01D0	add eax, edx		EDI 00000003
0345 A4	add eax, dword ptr ss:[ebp-5C]	Relocate Address = New_Imagebase + Offset + RVA	EIP 009B0D83 odbcad32.009B0D83
8985 5CFFFFFF	mov dword ptr ss:[ebp-A4], eax		C 0 ES 0023 32bit 0(FFFFFFFF)
89C7	mov edi, eax		P 0 CS 001B 32bit 0(FFFFFFFF)
8B35 28529B00	mov esi, dword ptr ds:[9B5228]		A 0 SS 0023 32bit 0(FFFFFFFF)
0335 50529B00	add esi, dword ptr ds:[9B5250]		

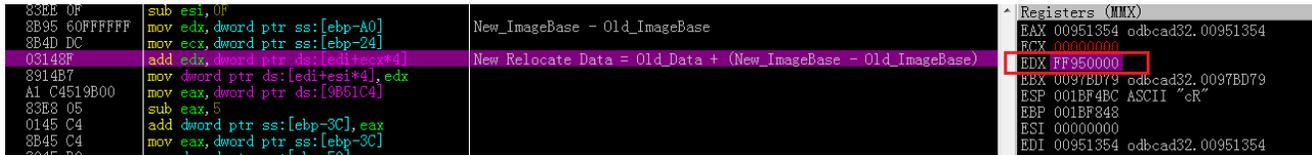
Figure 7: Calculation of the address to be relocated in Expiro's code.

As we see in the following screenshot, the formula leads to  $Relocation\_Address = 0x950000 + 0x354 + 0x1000$ , so the address in 0x951354 should be relocated (stored in eax).

A1 F4509B00	mov eax, dword ptr ds:[9B50F4]		Registers (MMX)
0305 F4519B00	add eax, dword ptr ds:[9B51F4]		EAX 00951354 odbcad32.00951354
83E8 06	sub eax, 6		ECX 00000000
0FB795 72FFFFFF	movzx edx, word ptr ss:[ebp-8E]		EDX 00000354
39D0	cmp eax, edx		EBX 0097BD79 odbcad32.0097BD79
75 32	jnz short 009B0DAC		ESP 001BF4BC ASCII "cR"
8B45 D0	mov eax, dword ptr ss:[ebp-30]	New_ImageBase	EBP 001BF848
0FB755 BA	movzx edx, word ptr ss:[ebp-46]	Offset	ESI 0000000B
01D0	add eax, edx		EDI 00000003
0345 A4	add eax, dword ptr ss:[ebp-5C]	Relocate Address = New_Imagebase + Offset + RVA	EIP 009B0D86 odbcad32.009B0D86
8985 5CFFFFFF	mov dword ptr ss:[ebp-A4], eax		C 0 ES 0023 32bit 0(FFFFFFFF)
89C7	mov edi, eax		P 0 CS 001B 32bit 0(FFFFFFFF)
8B35 28529B00	mov esi, dword ptr ds:[9B5228]		A 0 SS 0023 32bit 0(FFFFFFFF)
0335 50529B00	add esi, dword ptr ds:[9B5250]		

Figure 8: Relocation address being calculated.

In recovery step 3, Expiro computes the relocation value using the formula  $\text{Relocation\_Value} = \text{OldValue} + (\text{NewImageBase} - \text{OldImagebase})$ .



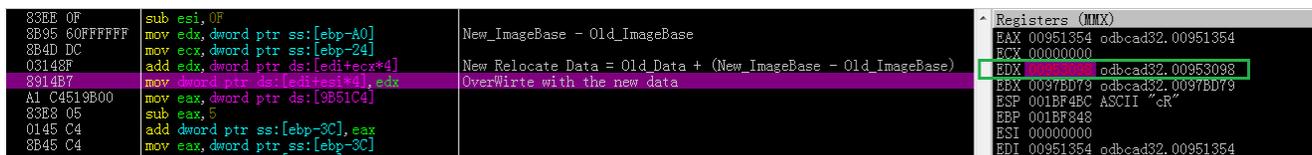
```
83EE 0F      sub esi, 0F
8B95 60FFFFFF mov edx, dword ptr ss:[ebp-A0]
8B4D DC      mov ecx, dword ptr ss:[ebp-24]
03148F      add edx, dword ptr ds:[edi+ecx*4]
8914B7      mov dword ptr ds:[edi+esi*4], edx
A1 C4519B00 mov eax, dword ptr ds:[9B51C4]
83E8 05      sub eax, 5
0145 C4      add dword ptr ss:[ebp-3C], eax
8B45 C4      mov eax, dword ptr ss:[ebp-3C]
8345 05      sub eax, 5
```

Registers (MMX)

EAX	00951354	odbcad32.00951354
ECX	00000000	
EDX	FF950000	
EBX	0097BD79	odbcad32.0097BD79
ESP	001BF4BC	ASCII "cr"
EBP	001BF848	
ESI	00000000	
EDI	00951354	odbcad32.00951354

Figure 9: Relocation value being computed by Expiro.

In this case, the formula is  $\text{Relocation\_Value} = 0x01001354 + (0x00950000 - 0x01000000)$ , so the relocation value is 0x00951354.



```
83EE 0F      sub esi, 0F
8B95 60FFFFFF mov edx, dword ptr ss:[ebp-A0]
8B4D DC      mov ecx, dword ptr ss:[ebp-24]
03148F      add edx, dword ptr ds:[edi+ecx*4]
8914B7      mov dword ptr ds:[edi+esi*4], edx
A1 C4519B00 mov eax, dword ptr ds:[9B51C4]
83E8 05      sub eax, 5
0145 C4      add dword ptr ss:[ebp-3C], eax
8B45 C4      mov eax, dword ptr ss:[ebp-3C]
8345 05      sub eax, 5
```

Registers (MMX)

EAX	00951354	odbcad32.00951354
ECX	00000000	
EDX	00953098	odbcad32.00953098
EBX	0097BD79	odbcad32.0097BD79
ESP	001BF4BC	ASCII "cr"
EBP	001BF848	
ESI	00000000	
EDI	00951354	odbcad32.00951354

Figure 10: Expiro performing relocations on its own.

Using this technique, we can decrypt and repair the entire relocation table of the files infected by Expiro. This also helps us to calculate and replace the relocation table size in an executable's optional header with the correct values. These changes ensure the infected files can run properly after removing the malicious payload.

McAfee products detect Expiro as W32/Expiro.gen.rd and W64/Expiro.d and repair infected files from DAT Version 8665. Users can find additional information at this [McAfee Labs Threat Advisory](#).

## SHA-256 hash

f15b8fc3ca117ab38e3074adc6208666b2189259e447db8202ef85b9bbfc4537

[Xiaobing Lin](#)

## More from McAfee Labs

[Crypto Scammers Exploit: Elon Musk Speaks on Cryptocurrency](#)

By Oliver Devane Update: In the past 24 hours (from time of publication) McAfee has identified 15...

May 05, 2022 | 4 MIN READ

[Instagram Credentials Stealer: Disguised as Mod App](#)

Authored by Dexter Shin McAfee's Mobile Research Team introduced a new Android malware targeting Instagram users who...

May 03, 2022 | 4 MIN READ

### Instagram Credentials Stealers: Free Followers or Free Likes

Authored by Dexter Shin Instagram has become a platform with over a billion monthly active users. Many...

May 03, 2022 | 6 MIN READ



### Scammers are Exploiting Ukraine Donations

Authored by Vallabh Chole and Oliver Devane Scammers are very quick at reacting to current events, so...

Apr 01, 2022 | 7 MIN READ



### Imposter Netflix Chrome Extension Dupes 100k Users

Authored by Oliver Devane, Vallabh Chole, and Aayush Tyagi McAfee has recently observed several malicious Chrome Extensions...

Mar 10, 2022 | 8 MIN READ



### Why Am I Getting All These Notifications on my Phone?

Authored by Oliver Devane and Vallabh Chole Notifications on Chrome and Edge, both desktop browsers, are commonplace,...

Feb 25, 2022 | 5 MIN READ



### Emotet's Uncommon Approach of Masking IP Addresses

In a recent campaign of Emotet, McAfee Researchers observed a change in techniques. The Emotet maldoc was...

Feb 04, 2022 | 4 MIN READ



### HANCITOR DOC drops via CLIPBOARD

Hancitor, a loader that provides Malware as a Service, has been observed distributing malware such as FickerStealer,...

Dec 13, 2021 | 6 MIN READ



### 'Tis the Season for Scams

## 'Tis the Season for Scams

Nov 29, 2021 | 18 MIN READ



### The Newest Malicious Actor: "Squirrelwaffle" Malicious Doc.

Authored By Kiran Raj Due to their widespread use, Office Documents are commonly used by Malicious actors...

Nov 10, 2021 | 4 MIN READ



### Social Network Account Stealers Hidden in Android Gaming Hacking Tool

Authored by: Wenfeng Yu McAfee Mobile Research team recently discovered a new piece of malware that specifically...

Oct 19, 2021 | 6 MIN READ



## Malicious PowerPoint Documents on the Rise

Authored by Anuradha M McAfee Labs have observed a new phishing campaign that utilizes macro capabilities available...

Sep 21, 2021 | 6 MIN READ

