

AutoIt Hata Ayıklaması

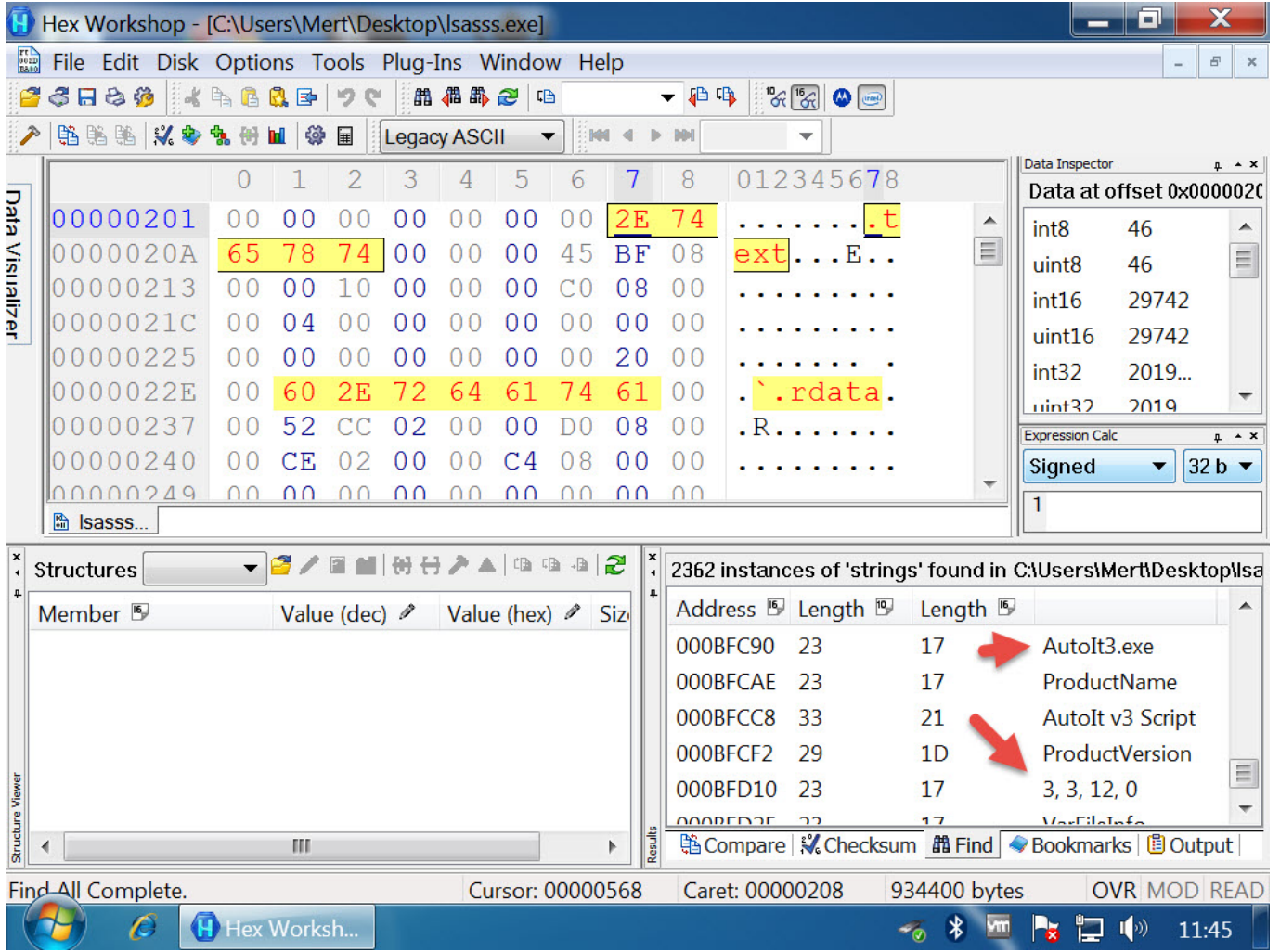
written by Mert SARICA | 1 February 2017

Hem Pi Hediye Var oyununun altıncısına hem de AutoIt Bankacılık Zararlı Yazılımı başlıklı blog yazıma konu olan zararlı yazılımı/betiği incelediğimizde, AutoIt'in son yıllarda zararlı yazılım geliştiriciler tarafından sıklıkla kullanıldığını görebiliyoruz. APT gibi hedeflenmiş siber saldırılarda da AutoIt ile geliştirilmiş zararlı yazılımların kullanılıyor olması, zararlı yazılım analistleri ve zararlı yazılım analizi becerisine sahip siber güvenlik uzmanları tarafından analiz edilebilmesini ihtiyaç haline getirmektedir.

Wikipedia'dan alıntı yapacak olursak, "AutoIt, Microsoft Windows için ücretsiz bir otomasyon yazılımıdır. Yazılımın ilk versiyonları tamamen otomasyona yönelik hazırlanmış olsa da sonradan kapsamı genişletilerek hemen her türlü uygulamanın geliştirilebileceği bir programlama aracı haline gelmiştir. Bir AutoIt betiği, AutoIt yorumlayıcısının yüklü olmadığı bilgisayarlarda çalışabilecek şekilde, sıkıştırılmış bir EXE programı haline getirilebilir."

Eğer "AutoIt Bankacılık Zararlı Yazılımı" başlıklı blog yazımdaki gibi şanslıysak, elimizdeki AutoIt betiğini (script) çeşitli hata ayıklama (debug) araçları ile analiz edebiliriz. Eğer derlenmiş, exe uzantılı bir AutoIt dosyası ile karşı karşıya isek bu durumda yapacağımız ilk iş, derlenmiş AutoIt dosyasını, betiğe çevirmek olacaktır.

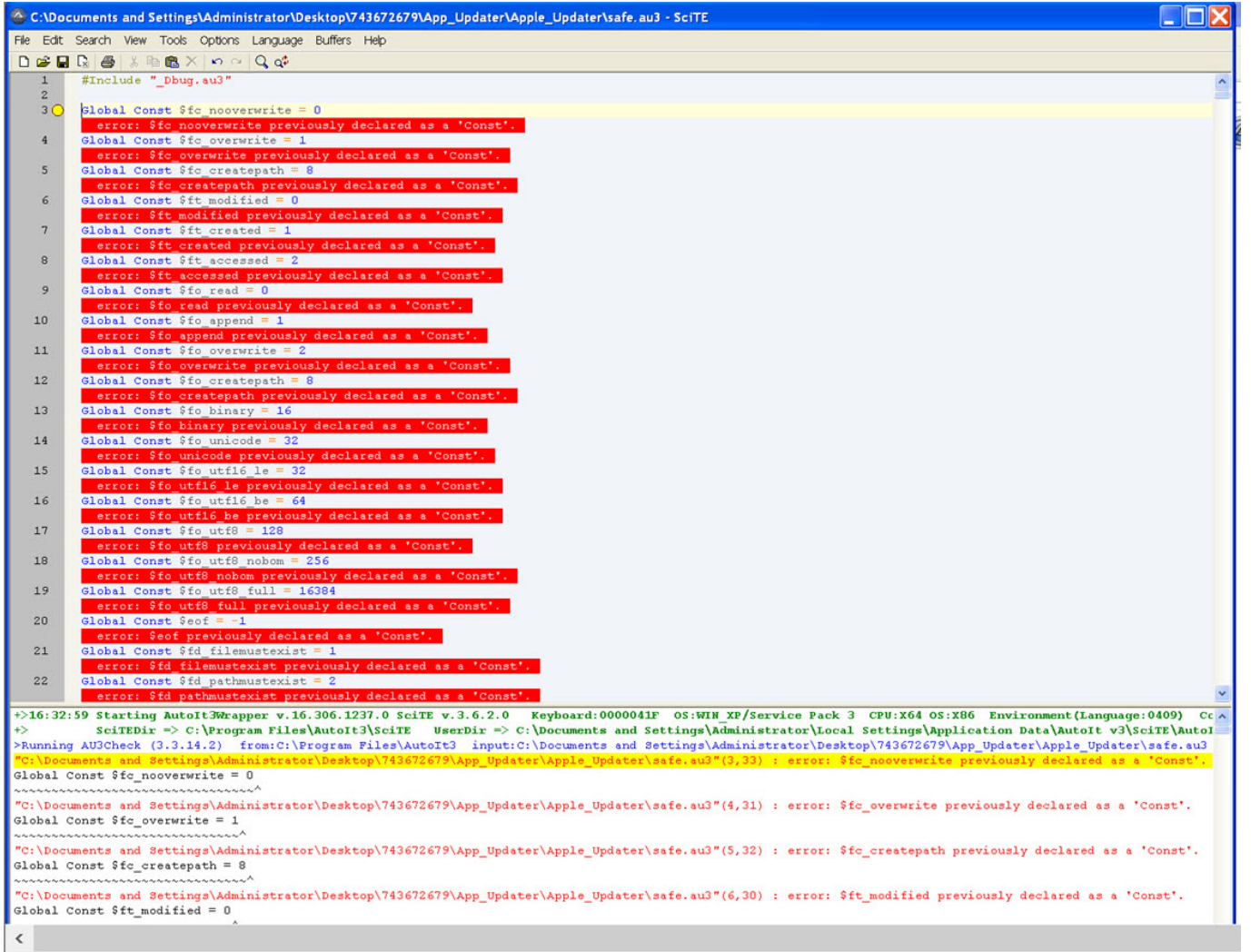
Derlenmiş AutoIt dosyasını betiğe çevirmek için Exe2aut aracından faydalanabiliriz. Exe2aut aracını çalıştırdıktan sonra exe uzantılı AutoIt dosyasını araca sürükledikten sonra betik dosyasına kolaylıkla ulaşabiliyoruz.



Hata ayıklama araçları (betikleri) arasında Dbug aracı, diğer araçlara kıyasla daha kullanışlı olduğu için onunla ilerleyebilirsiniz.

Dbug aracını kullanmak için öncelikle Auto IT script editörü olan SciTE aracını yüklememiz gerekiyor. Dbug aracının kurulum paketinden çıkan _Dbug.au3 dosyasını, analiz etmek istediğimiz safe betiği ile aynı klasöre koyduktan sonra safe betiğinin ilk satırına #Include "_Dbug.au3" satırını ekliyoruz. Bu işlemi gerçekleştirdikten sonra Dbug hata ayıklama aracını/betiğini çalıştırmak için başka bir eksikimiz kalmıyor.

safe betiğini SciTE ile açtıktan sonra ilk olarak F5 (run/resume execution) tuşuna basarak Dbug aracının devreye girmesini sağlıyoruz. Fakat betiği çalıştırdığımızda, AutoIt kütüphanesindeki değişkenler ve fonksiyonlar ile betikteki çakıştığı için soruna yol açan bu değişkenleri ve fonksiyonları silmemiz gerekiyor.



```
#Include "_Dbug.au3"

Global Const $fc_nooverwrite = 0
error: $fc_nooverwrite previously declared as a 'Const'.
Global Const $fc_overwrite = 1
error: $fc_overwrite previously declared as a 'Const'.
Global Const $fc_createpath = 8
error: $fc_createpath previously declared as a 'Const'.
Global Const $ft_modified = 0
error: $ft_modified previously declared as a 'Const'.
Global Const $ft_created = 1
error: $ft_created previously declared as a 'Const'.
Global Const $ft_accessed = 2
error: $ft_accessed previously declared as a 'Const'.
Global Const $fo_read = 0
error: $fo_read previously declared as a 'Const'.
Global Const $fo_append = 1
error: $fo_append previously declared as a 'Const'.
Global Const $fo_overwrite = 2
error: $fo_overwrite previously declared as a 'Const'.
Global Const $fo_createpath = 8
error: $fo_createpath previously declared as a 'Const'.
Global Const $fo_binary = 16
error: $fo_binary previously declared as a 'Const'.
Global Const $fo_unicode = 32
error: $fo_unicode previously declared as a 'Const'.
Global Const $fo_utf16_le = 32
error: $fo_utf16_le previously declared as a 'Const'.
Global Const $fo_utf16_be = 64
error: $fo_utf16_be previously declared as a 'Const'.
Global Const $fo_utf8 = 128
error: $fo_utf8 previously declared as a 'Const'.
Global Const $fo_utf8_nohom = 256
error: $fo_utf8_nohom previously declared as a 'Const'.
Global Const $fo_utf8_full = 16384
error: $fo_utf8_full previously declared as a 'Const'.
Global Const $eof = 1
error: $eof previously declared as a 'Const'.
Global Const $fd_filemustexist = 1
error: $fd_filemustexist previously declared as a 'Const'.
Global Const $fd_pathmustexist = 2
error: $fd_pathmustexist previously declared as a 'Const'.
```

+>16:32:59 Starting AutoIt3Wrapper v.16.306.1237.0 SciTE v.3.6.2.0 Keyboard:000041F OS:WIN XP/Service Pack 3 CPU:X64 OS:X86 Environment(Language:0409) Cc
+> SciTEDir => C:\Program Files\AutoIt3\SciTE UserDir => C:\Documents and Settings\Administrator\Local Settings\Application Data\AutoIt v3\SciTE\AutoI
>Running AU3Check (3.3.14.2) from:C:\Program Files\AutoIt3 input:C:\Documents and Settings\Administrator\Desktop\743672679\App_Updater\Apple_Updater\safe.au3
"C:\Documents and Settings\Administrator\Desktop\743672679\App_Updater\Apple_Updater\safe.au3"(3,33) : error: \$fc_nooverwrite previously declared as a 'Const'.
~~~~~  
"C:\Documents and Settings\Administrator\Desktop\743672679\App\_Updater\Apple\_Updater\safe.au3"(4,31) : error: \$fc\_overwrite previously declared as a 'Const'.  
~~~~~  
Global Const \$fc_overwrite = 1
~~~~~  
"C:\Documents and Settings\Administrator\Desktop\743672679\App\_Updater\Apple\_Updater\safe.au3"(5,32) : error: \$fc\_createpath previously declared as a 'Const'.  
~~~~~  
Global Const \$fc_createpath = 8
~~~~~  
"C:\Documents and Settings\Administrator\Desktop\743672679\App\_Updater\Apple\_Updater\safe.au3"(6,30) : error: \$ft\_modified previously declared as a 'Const'.  
~~~~~  
Global Const \$ft_modified = 0

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Betiği sorunsuz bir şekilde çalıştırdıktan sonra gözümüze kestirdiğimiz bir satırın üzerine gelip F9 (run to cursor) tuşuna basarak program akışının o satıra kadar ilerlemesini sağlıyoruz. 3711. satıra geldiğimizde farenin imlecini (cursor) bir üst satırda yer alan \$lc7065203373 değişkenin üzerine getirdiğimizde, o değişkenin hangi değere (<http://149.202.206.57>) sahip olduğunu görebiliyoruz. OllyDbg aracında olduğu gibi F7 tuşuna basarak ilgili fonksiyonun içine girebiliyor, F8 tuşuna basarak ise (step over) fonksiyonun içine girmeden akışın (flow) devam etmesini sağlayabiliyoruz. Özetle Dbug aracı sayesinde adım adım fonksiyonların ne işe yaradığını öğrenerek, fonksiyonların yanına yorum (comment) da yazarak kısa bir süre içinde zararlı yazılımın/betiğin ne iş yaptığını kolaylıkla öğrenebiliyoruz.

