Zararlı JavaScript Analizi

written by Mert SARICA | 1 November 2016

JavaScript, yaygın olarak internet tarayıcılarında kullanılan bir programlama dilidir. İnternet tarayıcılarında kullanılması nedeniyle de çoğunlukla güvenlik araştırmacıları ve art niyetli kişilerce internet tarayıcılarında güvenlik zafiyetleri tespit etmek ve tespit edilen güvenlik zafiyetlerini istismar etmek (Örnek: Aurora Operasyonu) amacıyla da kullanılmaktadır.

Ayrıca JavaScript'in istismar kitleri tarafından hedef sistemin kontrolünü ele geçirmek ve zararlı yazılım yüklemek amacıyla kullanıldığı da görülmektedir. Bir zamana damgasını vuran ve hem son kullanıcılara hem de kurumsal kullanıcılara bir hayli zor günler yaşatan Blackhole istismar kitini ve geliştiricisi tutuklandıktan sonra istismar kiti piyasasını ele geçiren Angler istismar kitini buna örnek gösterebiliriz.



Geçtiğimiz aylarda, Angler istismar kitinin ele geçirdiği hedef sistemlere Cryptowall zararlı yazılımı yüklediği, Heimdal güvenlik firması tarafından yapılan bir araştırma sonucunda ortaya çıktı.

Hem kurumlara gerçekleştirilen siber saldırılara hem de istismar kitlerine bakıldığında, JavaScript'in önemli bir role sahip olduğu görülüyor. Durum böyle olunca da, zararlı JavaScript kodlarının güvenlik uzmanları tarafından analiz edilebilmesi daha da önem kazanmaya başladı.

JavaScript'i OllyDbg, Immunity Debugger veya IDA Pro gibi hata ayıklayıcı araçları ile adım adım analiz etmek pek mümkün değil. Bunlar yerine JavaScript hata ayıklaması özelliğine sahip olan ve hem Chrome hem de Firefox internet tarayıcısı eklentisi olarak yüklenebilen, ücretsiz Firebug eklentisinden faydalanabilirsiniz.

Elinizde analiz etmeniz gereken ve Angler istismar kitine ait "trafik dosyası" (PCAP) olduğunu düşünelim. Yapacağınız ilk iş, http filtresi ve

Follow TCP Stream ile Wireshark aracında, HTML dosyası içinde yer alan şüpheli JavaScript'i tespit etmek olacaktır.



JavaScript'i başarıyla tespit ettikten sonra HTML kodlarından kurtulup dosya içinde sadece JavaScript kodlarını bırakmak, formatını biraz düzenlemek ve hata ayıklamaya başlayacağınız yere debugger; anahtar kelimesini yerleştirmek, analizinizi kolaylaştıracaktır. (debugger; anahtar kelimesini kullanmak yerine farklı yollar da izleyebilirsiniz.) Ancak Angler istismar kitinde olduğu gibi kimi durumlarda, JavaScript çalışma esnasında HTML kodlarına da ihtiyaç duyduğu için HTML kodlarından kurtulmak işinizi aksine zorlaştırabilir.



Hata ayıklamaya başlamak için Firefox internet tarayıcısını açıp, sağ üst köşede bulunan Firefox ikonuna bastığınızda, Firebug aktif hale geldiğini görebilirsiniz. Ardından Javascript kodunu içeren HTML dosyasını Firefox internet tarayıcısı ile açtığınızda, Firebug hata ayıklayıcısının debugger; anahtar kelimesi üzerinde sizden komut bekler halde beklediğini görebilirsiniz. OllyDbg, Immunity Debugger vb. hata ayıklayıcılar kullananıyorsunuz, aşağı bölümde, sağ üst köşede yer alan butonlar veya kısayollar (F8 – Continue, F11 – Step Into, F10 – Step Over) sayesinde FireBug'ı benzer şekilde kullanabilirsiniz.

Be Edit liver History Bookmarks Isola Belp	.6
Protex refers source and refers	ッ C Q Sardh 会自 手 合 々 ピ - 三
🥝 Disable+ 🛓 Cookies- 🎢 CSS- 🗋 Forms- 🖾 Images- 🕕 Information- 🚆 Miscellaneous- 🥖 Outline- 🥒 Resize- 🎇 Tools- 🔳 View Source	ter ll Oboue-
	C C C C C C C C C C C C C C
Porticada Dostructor Dostructor Porticada Dostructor	Omega Appendix Omega Appendix Operation Description Hetery Appendix Addrons Synce Colours Colours Packet Areunder's to go to file
System sources can't be debugged	Tarlo Danama Tarkan kakad

Hata ayıklamaya başladığınızda, Firebug aracının eval() fonksiyonu üzerinde durduğunu göreceksiniz. Sağ tarafta ise eval() fonksiyonu tarafından yorumlanacak/çalıştırılacak olan H4BA5 değişkeninde yer alan JavaScript kodlarını görebilirsiniz. Bu durum size, JavaScript kodlarının bu HTML kod içerisinde gizlenmiş olduğunu, internet tarayıcısı tarafından, çalışma (run) esnasında açık hale geldiğini (decode) ve çalıştırılmak üzere eval() fonksiyonuna iletildiğini açıkça göstermektedir.

Bile Edit Yew Higtory Bookmarks Tools Help		
🕐 Ber//C:/Docml_formathtml x 🔹 Online JavaScript beautifer 🛛 🗴 🎬 Free Online HTML Formatter x 🛛 🕂		
🗲 🛞 Re:///C:/Documents and Settings/Administrator/Desktop/js/angler_html_format.html	V X Q, Search	☆ 自 ♣ 舎 〃 ♥ - 三
🖉 Disable- 🛓 Cookies- 🧨 CSS- 😰 Forms- 🔤 Images- 🚯 Information- 🗧 Miscelaneous- 🥖 Outline- 🥜 Rester- 💥 Tools- 🔳 View Source- 🔝 Options-		
$ \begin{array}{c} 0.19 \\ 0.219 \\ 0.219 \\ 0.219 \\ 0.219 \\ 0.219 \\ 0.210 \\$	TO IF BIXX DI 18 BE 7 6 LA 08 Sould 24 SI FE DI D C 29 01 B 30 28 08 DB 51 69 14 31 CN D 70 141 89 D EB P 69 D 16 B F A3 80 FI P9 T0 60 38 E FALDA 12 B 5 DF BE C 6F F8C A 69 2 220 DEC 18 94L D C FI A3 AA BE 13 A 2 FI AF C 77T AF ED 52 DF 64 D 13 18 94L D C FI A3 AA BE 13 A 2 FI AF C 77T AF ED 52 DF 64 D 13 18 94L D C FI A3 AA BE 13 A 2 FI AF C 77T AF ED 52 DF 64 D 13 18 94L D C FI A3 AA BE 13 A 2 FI AF C 77T AF ED 52 DF 64 D 13 18 94L D C FI A3 AA BE 13 A 2 FI AF C 77T AF ED 52 DF 64 D 13 18 94L D C FI A3 AA BE 13 A 2 FI AF C 77T AF ED 52 DF 64 D 13 19 2 DF 04 D A 50 DF 70 FI 54 DF 04 FI A0 20 DF 70 DF 7	08.85 TB 31 CB 1 BG F TE3 16 ED 1 TE3T DRNS Q C 31 CC 33 0 BB 7 TEA. STP DRIT T3 15 G 06 11 SA D 11 6 A0 C 9 A 7 CD 87 89 FE D B D D A2 21 AF E1 5D 4 CB 04 FD 38C C B 7 3 E3 Q 60 KeN 17B E1 0 62 3 A 39 86 D 10 9 AD 2 CP 38 A 7 E × 0 B 45 C 28 FF 31 C A FE F 7 A 40 A 40 4 129 FE JD A DB C 118 3 A 21 80 4 0 2 A 0 E E A 16 A 31 2 A AD 9 KE A 129 FE JD A DB C 118 3 A 21 80 4 0 2 A 0 E E A 16 A 30 12 A AD 9 KE A 129 FE JD A DB C 118 3 A 21 80 4 0 2 A 0 E E A 16 A 30 12 A AD 9 KE A 129 FE JD A 20 C 118 3 A 21 80 4 0 2 A 0 E E A 16 0 A 17 0 1 0 0 0 FF 1 1 0 3 FD 10 7 120 FE JD A 10 C 118 3 A 21 80 4 0 2 A 0 E E A 10 9 6 FF 10 97 TB 36 60 DB 170 C TD 0 C C 31 0 9M 18 D 3 A C 17 0 9 1 0 0 FF 1 1 0 A 17 0 10 0 FF 1 1 0 A 17 0 7 170 A 20 A 27 A 27 1 1 0 F 9 D 0 1 E5 T 1 1 9 A 3 F F 9 F 1 1 C A FE D A1 CA C 6 TE F 22 S 41 3 E C 1 C TD 9 0 C C 2 5 0 F A AP B1 38 90 FE 0 0 9 E D 3 A 3 A 27 A 27 1 0 T 1 1 B FD 60 B 40 7 0 A 1 A 10 6 5 0 85 90 5 90 F - 0 3 T 38 38 A 6 9D C 0 9 BU 1 C A 5 D 1 4 B 7 0 A 3 1 A 0 5 C 3 8 9 (C 1 9 9 C C B C 0 0 S 15 0 A 4 13 4 2 B 60 T 9 FB 10 E 6 3 An 2 2 J 3 F 1 D 1 EF E H 2 D 1 T 7 RC 4 11 14 12 B 10 6 0 A An 2 2 J 3 F 1 0 1 FE E H 2 D 1 T 7 RC 4 11 14 14 10 B 60 A AN C 2 J 3 F 1 0 FE E H 2 B J 1 T 7 RC 4 11 14 14 10 B 60 A A 1 C 2 J 3 F 1 0 FE E H 2 B J 1 T 7 RC 4 11 14 1 2 B 6 8 0 F F C 1 0 9 5 9 O C F 6 C B C C 0 F 0 C 15 8 0 A A C 5 9 50 0 B C A 7 1 F 1 6 F 0 0 B 9 B 0 C C B C C 0 0 5 10 F 5 0 C F C C C B C 10 S 5 0 C F C C C 5 0 B 5 0 A F C 2 0 3 9 F 0 0 F 0 0 A 2 1 1 0 1 0 1 C C F E S 4 0 5 B 1 C C 17 C 0 9 C C C C C 1 5 0 B C C F C C 1 5 0 C C C C C 1 5 C B C C C C C 1 5 C B C C C C C C 1 5 0 C C C C C C C C C C C C C C C C C C
P T 1900 TH BALANT BALANT AND BALANT AND BALANT AND CALLED AND BALANT AND CALLED AND BALANT AND CALLED AND BALANT AND CALLED AND AND CALED AND CALLED AND CALLED AND CALLED AND	Puse # <number> to go to line</number>	
👪 al - angler_html_format.html - () - frRXAduEHmGCA.cd< < gtU < angler_htmat.html	🕨 🤽 🕄 况 🛛 Watch 🕶 Stack	Breakpoints
Clicking the 'Disable' button overrides 'debugger' with a disabled breakpoint.	Disable R New watch express	ion
<pre>'V to com databalanda brake notations on panel take menu: ''''''''''''''''''''''''''''''''''''</pre>	Elimitado	<pre>a Webser alps://by-is/statute</pre>
96 rbh(2) = 227(10,-ubstr(1, 4)); 97 rbh(2) = 227(10,-ubstr(1, 4)); 98 ukth umenesse ukth);	<u>~</u>	



Değişkenlerde yer alan JavaScript kodlarını dilediğiniz zaman konsol ekranından ulaşabilirsiniz. Bunun için Console ekranında console.log(değişken) yazmanız yeterlidir.



> console.log(h4BfA5);

Firebug eklentisi dışında, JavaScript kodlarını analiz etmek ve hata ayıkamak için Chrome internet tarayıcısı ile birlikte gelen Developer Tools aracından da faydalanabilirsiniz. Bunun için ilgili HTML dosyasını açtıktan sonra, klavyede F12 tuşuna basmanız yeterli olacaktır.

Diyelim ki elinizde yazının başında belirttiğim ve PCAP dosyasından çıkarttığınız, Angler istismar kitine ait bir HTML dosyası var ve bu dosyayı Chrome ile çalıştırdığınızda, internet tarayıcısının doodveeantoviar.kyderby.org adresine gitmeye çalıştığını farkettiniz. Bu adresin, HTML dosyasının tam olarak neresinde yer aldığını öğrenmek ve fonksiyonu analiz etmek istiyorsunuz ancak adresi arattınız ve bulamadınız, bu durumda ne yapacaksınız ?



Bu durumda yapmanız gereken ilk iş daha önce eval() fonksiyonu ile ortaya çıkan tüm JavaScript kodlarını, Angler HTML dosyasının içine kopyalamak ve çalıştırmak olacaktır. Çalıştırdığınız zaman aşağıdaki ekran görüntüsünde yer aldığı gibi http adresinin getKolaio() fonksiyonundan geldiğini görebilirsiniz. getKolaio() fonksiyonuna baktığınızda, bu fonksiyondan gelen değerin NhxUAmwU(xlAEe1xlAEe3); fonksiyonundan geldiğini görebilirsiniz. xlAEe1xlAEe3 değişkenine baktığınızda ise bunun gizlenmiş bir değere sahip olduğunu anlayabilirsiniz. NhxUAmwU fonksiyonunu incelediğinizde de, bunun gizlenmiş veriyi çözen ana fonksiyon olduğu ortaya çıkacaktır.





NhxUAmwU fonksiyonunu dinamik olarak hata ayıklayıcı ile analiz etmek için ise ilgili fonksiyonu ayrı bir HTML dosyasına kopyalayıp, gizlenmiş verileri bu fonksiyona ileterek, adım adım gizlenmiş verinin nasıl çözüldüğünü anlayabilir ve analizinizi gerçekleştirebilirsiniz. Yolunuz açık olsun :)

Ele Edit Vew Higtory Bookmarks IDools Help					
O file:///C:/Docop/decode.html × +					
e Re:///C:/Documents and Settings/Administrator/Desktop/decode.html		Search 🏠 🖨 🗍 🦸 🕈 🔿			
🖉 Disble - 1_ Cookles - 2/ CSS- [] Forms- 🔄 Images 🙃 Information- 📄 Miscelaneous - 2/ Outine - 2/ Respe- 💥 Tools- 🔳 View Source- []] Cookins-					
	-				
doodveeantoviar kyderby org					
	ОК				
🖋 🦞 🕻 >)= Console HTML CSS Script - DOM Net Cookies		PUse # <number> to go to line</number>			
Ik al • decode.html • {} Ik al • decode.html • {}	Watch - Stack B	reakpoints			
1 <html></html>	New watch expression	ha 🔨			
2 <script></script>					

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.