Botnet Zeus - analiza w laboratorium CERT

CERT Polska / NASK

11 stycznia 2011

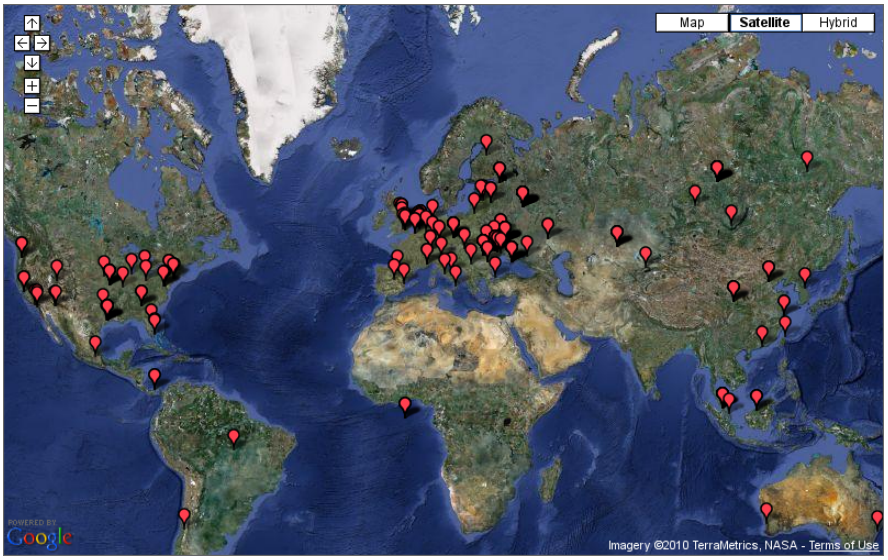
Streszczenie

W niniejszym opracowaniu opisane zostaną działania zespołu CERT Polska mające na celu analizę działania bota Zeus oraz opracowanie metody łamania zabezpieczeń transmisji pomiędzy botami a serwerami CC botnetu Zeus. Badania opierały się na analizie wstecznej zebranych próbek złośliwego oprogramowania. W wyniku badań powstała aplikacja umożliwiająca odszyfrowywanie i interpretację przechwyconych transmisji nowych wersji bota Zeus. Najpierw przedstawiona zostanie krótka charakterystyka botnetu Zeus i opisana zostanie komunikacja pomiędzy botem a serwerem CC. Następnie opisana zostanie przeprowadzona analiza bota, która doprowadziła do stworzenia narzędzi umożliwiających dekryptaż komunikacji.

1 Botnet Zeus

Zeus jest jednym z największych botnetów funkcjonujących obecnie w Internecie. Znany jest także pod nazwami: Zbot, PRG, Wsnpoem, Gorhax, Kneber. Boty Zeusa są koniami trojańskimi, których głównymi zadaniami są: kradzież danych dotyczących kont bankowych (przez przechwytywanie danych przesyłanych z klawiatury) oraz prowadzenie kampanii phishingowych. Wielkość botnetu Zeus szacuje się na kilkanaście milionów zainfekowanych komputerów, z czego 3.6 milionów znajduje się w USA [1]. Ofiarą botnetu padły m. in. przedsiębiorstwa: Bank of America, NASA, Monster, ABC, Oracle, Cisco, Amazon oraz BusinessWeek. Do 29. października 2009 roku botnet wysłał ponad 1.5 milionów wiadomości phishingowych do serwisu Facebook. W dniach 14–15 listopada 2009 botnet wysłał około 9 milionów sfałszowanych wiadomości podszywając się pod firmę Verizon Wireless [2]. W dniu 14. czerwca 2010 roku firma Trustee udostępniła raport na temat skompromitowanych z pomocą Zuesa kart kredytowych piętnastu amerykańskich banków [3]. W dniu 1. października

2010 roku FBI poinformowało o istnieniu międzynarodowej grupy cyberprzestępczej odpowiedzialnej za włamania do amerykańskich komputerów i kradzieży około 70 milionów dolarów. W USA aresztowano 90 osób, aresztowania nastąpiły także w Wielkiej Brytanii i na Ukrainie [4].



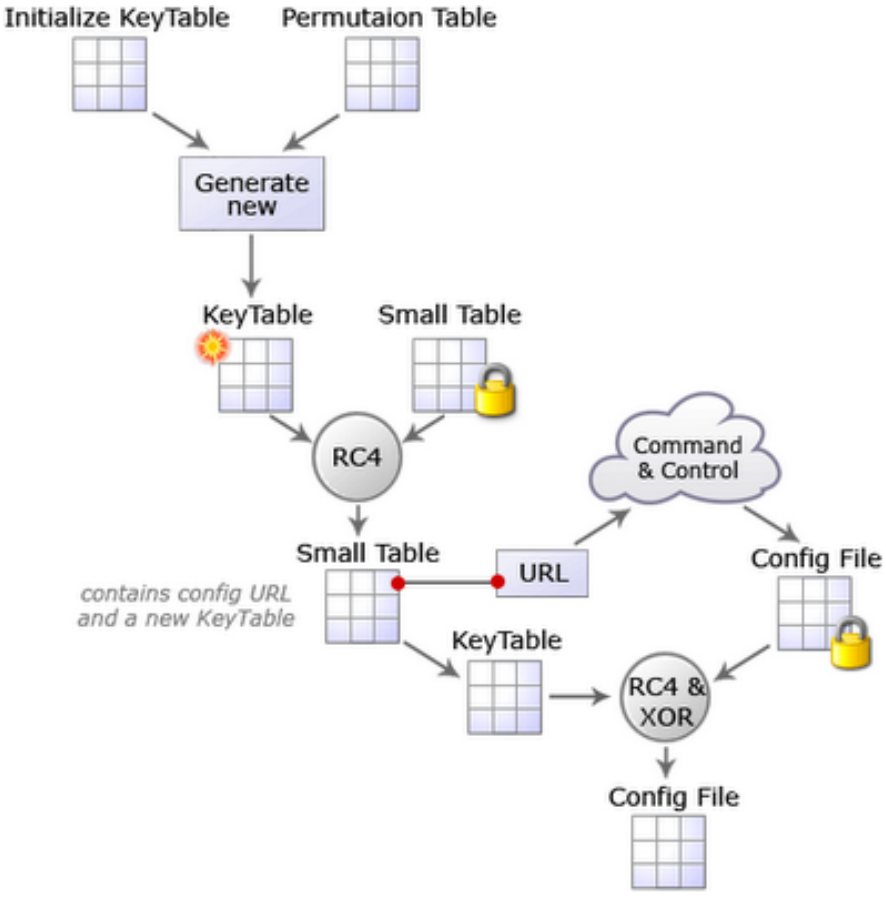
Mapa działających serwerów CC Zeusa (źródło: Zeus Tracker)

2 Bot Zeus

Boty Zeus (również: "Zboty") infekują komputery działające pod kontrolą systemu Windows. W przeciwieństwie do innych botów (Storm, Conficker), Zboty nie posiadają wbudowanego w siebie mechanizmu dalszej propagacji. Infekcje następują zazwyczaj w wyniku działania technik typu drive-by-download i kampanii spamersko-phishingowych, a nie poprzez automatyczne wyszukiwanie innych podatnych systemów. Infekcja Zbotem przebiega następująco: po uruchomieniu droppera (programu instalującego trojana) do procesu winlogon.exe (starsze wersje) lub explorer.exe (nowsze wersje) wstrzykiwany jest wątek zawierający złośliwy kod. Wątek ten pobiera z serwera CC plik konfiguracyjny (dalej nazywany "configiem") zawierający listę atakowanych banków, stron, wykorzystywane wektory ataków, etc.. Po załadowaniu informacji z configu Zbot przechwytuje krytyczne dla bezpieczeństwa użytkowników informacje (głównie informacje logowania) i okresowo wysyła je do serwera CC.

3 Komunikacja bot – CC

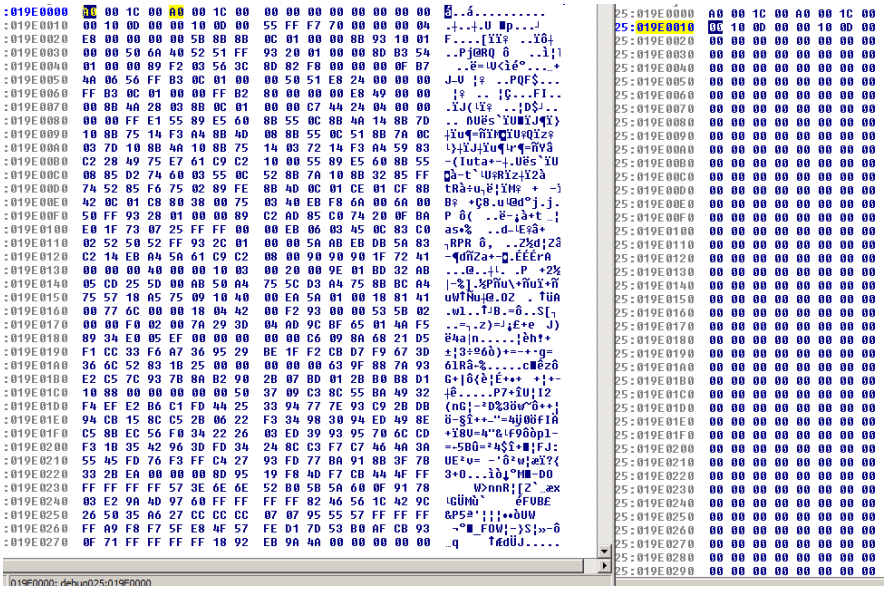
Aby utrudnić analizę bota, twórcy Zeusa wprowadzili do niego szereg zabezpieczeń wykorzystujących techniki kryptograficzne i obfuskacyjne, jak również polimorficzny kod. Komunikacja Zbota z serwerem CC jest szyfrowana za pomocą algorytmu RC4 (więcej informacji na temat algorytmu i kryptograficznych technik wykorzystywanych przez Zeus znajduje się w punkcie 8. niniejszego opracowania). Klucz wykorzystywany do szyfrowania transmisji jest tworzony na podstawie danych zawartych w każdej próbce Zbota. Twórcy Zeusa zastąpili oryginalny algorytm konstruowania klucza RC4 własnym. Jego starsza wersja przedstawiona jest na ilustracji 2. Zespół CERT przeanalizował nową wersję Zbota, wstrzykującą wątek do procesu explorer.exe oraz poszerzającą algorytm konstrukcji klucza o pętlę xor.



Konstrukcja klucza (źródło: http://blog.threatexpert.com)

4 Analiza nowej wersji – dropper

Dropper Zeusa posiada kilka warstw zabezpieczeń utrudniających jego analizę. Proces ich przełamywania opisany został poniżej. Po uruchomieniu, dropper alokuje fragment pamięci operacyjnej, do którego następnie z różnych miejsc z pamięci przenosi dane. Po połączeniu tych fragmentów w jedną całość i odszyfrowaniu ich otrzymuje właściwy kod, który zaczyna wykonywać.



Zaalokowany fragment pamięci z deobfuskowanym kodem

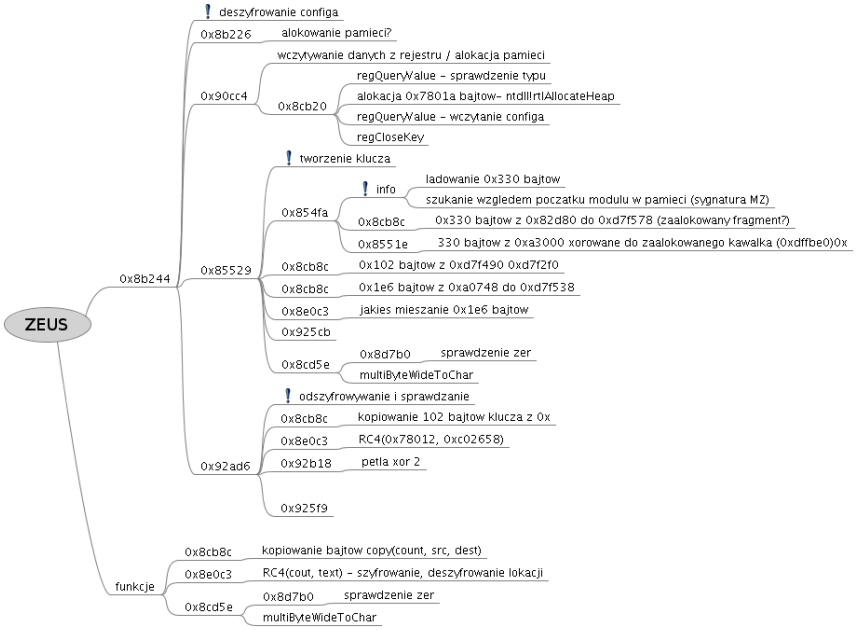
Kod ów za pomocą wywołań systemowych alokuje kolejny fragment pamięci, do którego ładuje złośliwy kod, omija system zabezpieczeń systemu Windows oparty na tzw. tokenach bezpieczeństwa i wstrzykuje skonstruowany wątek do procesu explorer.exe. Za każdym razem, kiedy system zostanie uruchomiony, a co za tym idzie, rozpocznie się wykonywanie programu explorer.exe, bot rozpocznie próby nawiązania komunikacji z serwerem CC.

5 Analiza nowej wersji – wstrzyknięty wątek

Dokładnej analizie poddany został fragment wątku procesu explorer.exe odpowiedzialny za pobranie configu z serwera CC. Uproszczony schemat działania przedstawiony został na ilustracji 4. Składa się na niego:

1. Wczytanie configu z klucza rejestru systemowego.
2. Konstrukcja klucza.
3. Odszyfrowanie za pomocą klucza oraz sprawdzenie skrótu md5 configu.

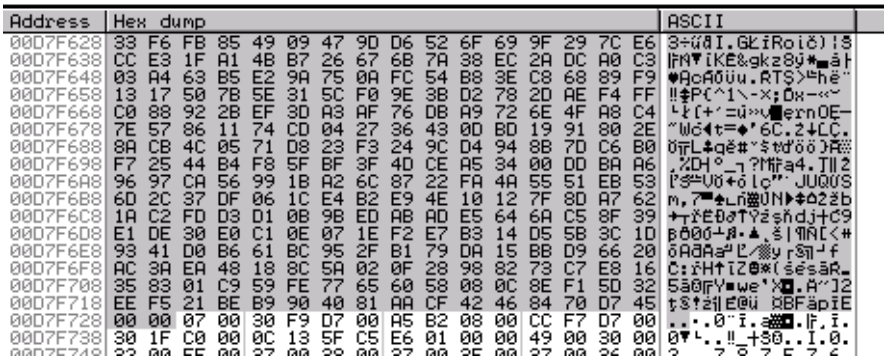
W trakcie kostruowania klucza (ilustracja 2) wykorzystywany jest algorytm RC4 oraz pętla xor. Po otrzymaniu klucza config jest odszyfrowywany algorytmem RC4 i pętlą xor.



Schemat odszyfrowywania configu

6 Deszyfrowanie komunikacji

Po wykonaniu zrzutu pamięci zawierającego szyfrogram configu i klucz możliwe jest odzyskanie tekstu jawnego. Do zautomatyzowania tego procesu stworzony został program decode. Program ten przeprowadza proces dekryptażu RC4 wykorzystując uzyskane klucze. Otrzymany, jawny config można poddać analizie za pomocą kolejnego narzędzia. W analizowanych configach znaleźliśmy m. in. instrukcje atakowania serwisu facebook, innych serwisów społecznościowych, serwisów firmy Microsoft, hiszpańskich oraz niemieckich banków i przedsiębiorstw. W pracowni CERT Polska został także opracowany program umożliwiający automatyczne odzyskiwanie szyfrogramów i kluczy na zainfekowanych komputerach.



Address	Hex dump	ASCII
00D7F628	33 F6 FB 85 49 09 47 9D D6 52 6F 69 9F 29 7C E6	3÷ûàI.GÖRoiƒ)|æ
00D7F638	CC E3 1F A1 4B B7 26 67 6B 7A 38 EC 2A DC A0 C3	Ìã¡K·&gkz8ì*Ü Ã
00D7F648	03 A4 63 B5 E2 9A 75 0A FC 54 B8 3E C8 68 89 F9	¤cµâ.uüT¸>Èh‰ù
00D7F658	13 17 50 7B 5E 31 5C F0 9E 3B D2 78 2D AE F4 FF	P{^1\ð;Òx-®ôÿ
00D7F668	C0 88 92 2B EF 3D A3 AF 76 DB A9 72 6E 4F A8 C4	À.'+ï=£¯vÛ©rnO¨Ä
00D7F678	7E 57 86 11 74 CD 04 27 36 43 0D BD 19 91 80 2E	~W.tÍ.'6C.½.'€.
00D7F688	8A CB 4C 05 71 D8 23 F3 24 9C D4 94 8B 7D C6 B0	ŠËL.qØ#ó$œÔ.‹}Æ°
00D7F698	F7 25 44 B4 F8 5F BF 3F 4D CE A5 34 00 DD BA A6	÷%D´ø_¿?MΥ4.Ýº¦
00D7F6A8	96 97 CA 56 99 1B A2 6C 87 22 FA 4A 55 51 EB 53	–—ÊV™.¢l‡"úJUQëS
00D7F6B8	6D 2C 37 DF 06 1C E4 B2 E9 4E 10 12 7F 8D A7 62	m,7ß..ä²éN.....§b
00D7F6C8	1A C2 FD D3 D1 0B 9B ED AB AD E5 64 6A C5 8F 39	.ÂýÓÑ.›í«.ådjÅ.9
00D7F6D8	E1 DE 30 E0 C1 0E 07 1E F2 E7 B3 14 D5 5B 3C 1D	áÞ0àÁ.....òç³.Õ[<.
00D7F6E8	93 41 D0 B6 61 BC 95 2F B1 79 DA 15 BB D9 66 20	"AÐ¶a¼•/±yÚ.»Ùf
00D7F6F8	AC 3A EA 48 18 8C 5A 02 0F 28 98 82 73 C7 E8 16	¬:êH.ŒZ..(˜‚sÇè.
00D7F708	35 83 01 C9 59 FE 77 65 60 58 08 0C 8E F1 5D 32	5ƒ.ÉYþwe`X..Žñ]2
00D7F718	EE F5 21 BE B9 90 40 81 AA CF 42 46 84 70 D7 45	îõ!¾¹.@.ªÏBF„p×E
00D7F728	00 00 07 00 30 F9 D7 00 A5 B2 08 00 CC F7 D7 00	....0ù×.¥²..Ì÷×.
00D7F738	30 1F C0 00 0C 13 5F C5 E6 01 00 00 49 00 30 00	0.À..._Åæ...I.0.

Klucz Zeusa w pamięci procesu explorer.exe

```
john@raticate ~/analysis/deco/decoder_v2 $ ./decode config1.dump config_key1.dump output1.dump
Input: config1.dump, 0x78012 bytes
Key: config_key1.dump, 0x102 bytes
```

Wywołanie programu decode

```
 |  Rec  size:     636 B ( 0000027C )   |
 |  Data size:    1275 B ( 000004FB )   |
 *  DATA @ >>
@https://[redacted]/Transactional/clientes/access/* 0x00
@https://[redacted]/publico/contents/* 0x00
@https://[redacted]/BEWeb/2031/2031/inicio_identificacion.action* 0x00
@https://[redacted]/home/acceso.asp* 0x00
@https://[redacted]/BEWeb/2105/* 0x00
@https://[redacted]/NASApp/ceconline/flow.jsp* 0x00
@https://[redacted]/ISMC/Manlleu_cat/acceso.jsp* 0x00
@https://[redacted]/BEWeb/2097/2097/inicio_identificacion.action* 0x00
@https://[redacted]/ibercaja/asp/Clave.asp* 0x00
@https://[redacted]/tx0001/TecladoVirtual.jsp* 0x00
@https://[redacted]/BEWeb/2045/6045/inicio_identificacion.action* 0x00
@https://[redacted]/CajaMadrid/oi/pt_oi/Login/login* 0x00
@https://[redacted]* 0x00
!"http://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
!"https://[redacted]/*" 0x00
 0x00
 << DATA

 .--------------------------------------.
 |  ID: 0x00000001   TYPE:0x40000001    |
 |  Rec  size:    8210 B ( 00002012 )   |
 |  Data size:   22789 B ( 00005905 )   |
 *  DATA @ >>
[0b] 0x00
 0x00
 0x00
</body>[0b] 0x00
 0x00
 0x00
</html>[ef]X 0x00
 0x00
 \r \n
<script id="injScp" type="text/javascript" language="JavaScript"> \r \n
```

Fragment odszyfrowanego configa

7 Algorytm RC4

Algorytm RC4 jest strumieniowym, symetrycznym algorytmem szyfrującym. Jego twórcą jest Ron Rivest. RC4 znajduje zastosowanie m.in. w algorytmach i protokołach WEP, SSL, BitTorrent. Aktualna wiedza nt. tego algorytmu nie pozwala na odzyskanie tekstu jawnego bez znajomości klucza na podstawie danych, którymi dysponują analitycy [6]. Ze względu na olbrzymią przestrzeń kluczy niemożliwe jest również jej przeszukanie. Płyną stąd następujące wnioski: wśród skutecznych metod analizy szyfrowanej komunikacji botnetu Zeus można wskazać: zrzut pamięci zawierającej tajny klucz i deszyfrowanie transmisji oraz ewentualnie analizę algorytmu generowania kluczy tajnych (o ile taki istnieje) i próby pomniejszenia przestrzeni kluczy. W trakcie prac nad botem wybraliśmy pierwszą z tych metod, jako najskuteczniejszą i najbardziej efektywną w czasie prowadzenia badań.

8 Podsumowanie

W efekcie analizy wstecznej próbek złośliwego oprogramowania Zeus w jednostce CERT Polska opracowano zestaw programów pozwalających na skuteczne deszyfrowanie transmisji botów z serwerami CC oraz przesyłanych configów. Otrzymane informacje pozwalają na reagowanie na przypadki odnalezienia nowych sieci botów atakujących nowe cele, analizę zachowania się botnetów oraz ostrzeganie zagrożonych instytucji i przedsiębiorstw.

Literatura

[1] *UAB computer forensics links internet postcards to virus.* The Hindu. July 27, 2009. Retrieved November 17, 2009.

[2] *New Verizon Wireless-themed Zeus campaign hits.* SC Magazine. November 16, 2009. Retrieved November 17, 2009.

[3] *Zeus/Zbot Trojan Attacks Credit Cards of 15 US Banks.* TechPP. Retrieved July 15, 2010.

[4] *CYBER BANKING FRAUD Global Partnerships Lead to Major Arrests.* FBI (October 1, 2010), Retrieved October 2, 2010.

[5] *A summary of the ZeuS Bot.* Richard S. Westmoreland

[6] *Analysis of the Stream Cipher RC4.* Itsik Mantin