Groupe de travail Réseau
Request for Comments : 3607
Catégorie : Information

M. Leech, Nortel Networks
septembre 2003
Traduction Claude Brière de L'Isle

Réexamen de l'analyse cryptographique dite Loterie chinoise : l'Internet comme outil de cassage de code

Résumé
Le présent document revoit l'attaque d'analyse de chiffrement massivement parallèle qu'on appelle la loterie chinoise. Il explore les attaques analogues fondées sur l'Internet à la loterie chinoise, et leurs conséquences potentiellement sérieuses.

1. Introduction

En 1991, Quisquater et Desmedt [DESMEDT91] ont proposé une attaque ésotérique, mais fondée techniquement, contre DES ou les chiffrements similaires. Ils appelaient cette attaque la loterie chinoise. Elle se fondait sur une approche de matériels massivement parallèles, en utilisant les appareils électroniques des consommateurs tels que les "hôtes" du matériel casseur de chiffre.

Dans la décade écoulée depuis que Quisquater et Desmedt ont proposé leur loterie chinoise à l'expérimentation, il y a eu une croissance considérable du nombre de domaines qui valent la peine qu'on revienne sur cette expérience.

En 1991, l'Internet avait approximativement 8 millions d'hôtes joignables rattachés et en 2002, le nombre est passé à l'incroyable 100 millions d'hôtes joignables. Depuis l'article sur la loterie chinoise, la puissance de calcul disponible pour un micro ordinateur d'utilisateur moyen a crû d'un facteur d'environ 150.

2. Synergie dangereuse

La croissance combinée de l'Internet et l'irrésistible marche de la loi de Moore se sont combinées pour créer un potentiel dangereux d'analyse cryptographique pour les systèmes de chiffrement existants.

Dans les quelques dernières années, plusieurs attaques de grande ampleur par ce qu'on appelle des vers de l'Internet [SPAFF91] ont réussi à compromettre et infecter un nombre étonnamment grand d'hôtes rattachés à l'Internet. En 2001, l'Association coopérative pour l'analyse des données de l'Internet [CAIDA2001] rapportait que le ver Code Red v2 était capable d'infecter plus de 350 000 hôtes dans ses 14 premières heures de fonctionnement. La charge utile du ver Code Red était une méchanceté : le site web de l'hôte était mutilé par un message politique. Il était hardi, effronté, et attirait presque immédiatement l'attention.

Considérons un moment un ver Internet avec des intentions plus sombres et en fin de compte plus dangereuses : analyser cryptographiquement en force brute un message, afin de déterminer la clé utilisée avec ce message. Pour que le ver réussisse, il doit éviter la détection pendant assez longtemps pour construire un niveau significatif de systèmes infectés, afin d'avoir suffisamment de cycles de CPU agrégés pour achever l'analyse cryptographique. De plus, notre ver devrait éviter la détection pendant assez longtemps pour que la clé cassée puisse être utile aux propriétaires du ver. Des recherches récentes [USEN2002] sur des vers furtifs peignent un tableau très sombre.

Même après qu'un tel ver ait été détecté, il serait presque impossible de dire quelle clé attaquait le ver. Toute charge utile d'attaque réaliste aura un ou deux morceaux de texte chiffré, et un peu de libellé connu, ou des caractéristiques de libellé probable associées ; sûrement pas assez de données pour déterminer la victime vraisemblable.

3. Le gagnant téléphone à la maison

Lorsque une instance donnée du ver détermine la clé, elle a besoin de contacter l'origine afin de lui donner la clé. Elle doit le faire de façon à minimiser la probabilité que l'origine se fasse prendre.

Une de ces techniques serait que le ver chiffre la clé en clé publique, sous la ou les clés publiques de la ou des origines, et la place dans un site anodin sur le site de la toile de l'hôte compromis. Le ver pourrait aussi la propager vers l'arrière de façon à ce qu'un certain nombre de sites compromis de la toile dans le voisinage topologique du ver contiennent aussi les données. Le fichier contenant la clé serait identifié avec un mot clé unique que les générateurs vont occasionnellement chercher avec des moteurs de recherche de l'Internet. Le ver pourrait rendre le processus plus efficace en utilisant les services de "registre de mots clés" de divers services de recherche de l'Internet.

Une autre approche serait d'envoyer un message (éventuellement chiffré en PGP) à plusieurs groupes de nouvelles, à travers un service d'envoi anonyme. De même, les services Internet de "chat" comme IRC pourraient être utilisés : il y a en fait une tradition émergente d'utilisation d'IRC et de services similaires pour le contrôle des vers et virus en temps réel, de façon anonyme.

N'importe laquelle de ces méthodes peut être utilisée à la fois pour permettre à l'instance de ver "gagnante" d'envoyer les résultats et pour que l'origine du ver envoie un nouveau travail à l'armée accumulée de systèmes compromis.

4. Évaluation de la menace

La croissance de l'Internet et des performance de CPU suivent toutes deux un intervalle de doublement raisonnablement prévisible. Les performances du matériel de calcul paraissent toujours suivre la loir de Moore, dans laquelle les performances doublent tous les 1,5 ans. La croissance de l'Internet apparaît comme suivant une période de doublement de trois ans.

En établissant une période commune pour les performances et la croissance de l'Internet, on peut facilement déterminer le temps d'attaque vraisemblable pour une année donnée, sur la base d'une approche purement arithmétique.

Une hypothèse simplificatrice est qu'il est bien sûr possible de construire un ver furtif à souhait et qu'il peut réaliser une infection en régime permanent d'environ 0,5 % de tous les hôtes rattachés à l'Internet.

En 1995, J. Touch, de l'ISI, a publié une analyse détaillée des performances de MD5 [RFC1810]. À cette époque la performance logicielle de MD5 culminait à 87 Mbit/s, soit un équivalent de 170 000 opérations MD5 d'un seul bloc par seconde. La même année, les performances de DES culminaient à environ 50 000 opérations setkey/encrypt par seconde.
En 1995, l'Internet avait environ 20 000 000 d'hôtes rattachés. En 2002, il y a maintenant 100 000 000 d'hôtes rattachés.

Un simple programme C, donné en Appendice A peut être utilisé pour prédire les performances de notre ver hypothétique pour une année donnée. Si on fait tourner ce programme pour 2002, cela donne :

Année d'estimation : 2002
Pour un nombre total d'hôtes infectés de 503 968
Performance agrégée : MD5 9,79e+11/s ; DES 2,88e+11/s
DES pourrait être cassé en environ 1,45 jours
DES avec des mots de passe de 8 caractères pourrait être cassé en 16,29 minutes
MD5 avec des clés de 64 bits pourrait être cassé en environ 218,04 jours
MD5 avec des mots de passe de 8 caractères pourrait être cassé en 4,79 minutes

Les nombres donnés ci-dessus suggèrent qu'une attaque indétectée contre MD5, pour une longueur de clé raisonnable, n'est pas vraisemblable en 2002. Une attaque réussie contre DES paraît cependant être une presque certitude.

5. Considérations pour la sécurité

La faiblesse de DES a été démontrée dans un passé récent. Le succès de la machine EFF, décrit dans [EFF98] montre comment un effort de matériel massivement parallèle peut réussir de façon relativement économique. Que ce niveau de force d'analyse cryptographique à force ouverte puisse devenir disponible à partir des matériels du commerce est une pensée raisonnable. Il est clair que DES doit être abandonné, en faveur de 3DES ou du plus récent AES [FIPS197].

Le tableau est moins sombre pour MD5 (lorsqu'il est utilisé dans de simple constructions MAC). Pour les messages courts les clés longues avec MD5 sont effectivement libres, du point de vue du calcul, de sorte qu'il paraît sensé d'utiliser les plus grandes longueurs de clé architecturalement praticables avec MD5.

Les logiciels de systèmes d'exploitation deviennent de plus en plus complexes et les concepteurs de vers, virus, chevaux de Trois de l'Internet, et autres malgiciels, deviennent plus sophistiqués. Bien que leur but ait largement été le vandalisme à grande échelle, il semble raisonnable de supposer que leurs méthodes pourraient être tournées vers une activité plus ciblée et peut être plus inquiétante.

En février 2003, au moins un ver, connu sous le nom de W32/Opaserv.A avait une charge utile destinée à mettre en œuvre un casseur de DES réparti.

6. Remerciements

John Morris, de Nortel IS, a contribué à l'idée d'un envoi par groupe de nouvelles anonyme.

Appendice A Code source

```
/*
* Ce programme calcule les performances d'un hypothétique ver d'analyse cryptographique "Loterie chinoise" à force ouverte, fondé sur la date actuelle, l'estimation du taux de croissance de l'Internet et de la loi de Moore.
*
*/ #include <stdio.h> #include <math.h> /*
* EPOCH pour les calculs
*/ #define EPOCH 1995.0 /*
* Taille de l'Internet (ca 1995)
*/ #define INTERNET_SIZE 20000000.0

/*
* Performances du logiciel MD5 (ca 1995)
*/ #define MD5PERF 170000.0

/*
* Performances du logiciel DES (ca 1995)
*/ #define DESPERF 50000.0

main (argc, argv) int argc; char **argv; {
        double yeardiff;
        double cryptoperf, multiplier;
        double cracktime;

        yeardiff = (double)atoi(argv[1]) - EPOCH;

        /*
         * L'intervalle de doublage des performances de la loi de Moore est de 1,5 ans
         */
        cryptoperf = yeardiff / 1.5;
        cryptoperf = pow(2.0, cryptoperf);

        /*
         * Un peu de confusion ici : tous les hôtes ne sont pas au plus rapide
         */
        cryptoperf *= 0.450;

        /*
         * L'intervalle de doublement de taille de l'Internet est tous les trois ans
         */
        multiplier = yeardiff / 3.0;
        multiplier = pow(2.0, multiplier);
        multiplier *= (INTERNET_SIZE*0.0050);
```

```
 fprintf (stderr, "Année d'estimation : %d\n", atoi(argv[1]));

 fprintf (stdout, "Pour un nombre total d'hôtes infectés de %d\n",
      (int)multiplier);
fprintf (stdout, "performance agrégée : MD5 %5.2e/sec DES %5.2e/sec\n",
    MD5PERF*cryptoperf*multiplier,
    DESPERF*cryptoperf*multiplier);

cracktime = pow(2.0, 55.0);
cracktime /= (DESPERF*cryptoperf*multiplier);
fprintf (stdout,
    "DES pourrait être cassé en environ %3.2lf days\n", cracktime/86400.0);

cracktime = pow(2.0, 8.0*6.0);
cracktime /= (DESPERF*cryptoperf*multiplier);
fprintf (stdout,
    "DES avec des mots de passe de 8 caractères pourrait être cassé en  %3.2lf minutes\n",cracktime/60);

cracktime = pow(2.0, 64.0);
cracktime /= (MD5PERF*cryptoperf*multiplier);
fprintf (stdout,
    "MD5 avec des clés de 64 bits pourrait être cassé en environ %3.2lf days\n", cracktime/86400.0);

cracktime = pow(2.0, 8.0*6.0);
cracktime /= (MD5PERF*cryptoperf*multiplier);
fprintf (stdout,
    "MD5 avec des mots de passe de 8 caractères pourrait être cassé en %3.2lf minutes\n", cracktime/60); }
```

Références normatives

[DESMEDT91] J. Quisquater, Y. Desmedt, "Chinese Lotto as an Exhaustive Code-Breaking Machine". Computer, v. 24, n. 11, Nov 1991, pp. 14-22.

[RFC1810] J. Touch, "Rapport sur les performances de MD5", RFC 1810, juin 1995.

[EFF98] "Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design", Electronic Frontier Foundation, 1998.

[CAIDA2001] "CAIDA Analysis of Code Red" http://www.caida.org/analysis/security/code-red/

[SPAFF91] "The Internet Worm Program: An Analysis", Eugene Spafford, 1991.

[FIPS197] "Advanced Encryption Standard", US FIPS197, November, 2001.

[USEN2002] "How to own the Internet in Your Spare Time", Proc. 11th Usenix Security Symposium, 2002.

Adresse de l'auteur

Marcus D. Leech
Nortel Networks
P.O. Box 3511, Station C
Ottawa, ON
Canada, K1Y 4H7
téléphone : +1 613-763-9145
mél : mleech@nortelnetworks.com

Déclaration complète de copyright

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.