

Compte-rendu Usenix Security Symposium 03 Washington, DC 4-8 août 2003

Nicolas Jombart

<Nicolas.Jombart@hsc.fr>

15 septembre 2003

Introduction

La douzième conférence Usenix Security Symposium, présentant traditionnellement les dernières avancées de la recherche en sécurité informatique, s'est déroulée début août à Washington, DC. La programmation a cette année été le plus à l'honneur, avec nombre d'articles présentant des méthodes de détection ou de protection contre les failles classiques de programmation, en particulier les débordements de tampon. Cependant, un sujet intéressant fortement les américains a lui aussi largement eu sa place, le vote électronique, utilisé récemment aux Etats-Unis.

La semaine commence par deux tutoriels (quatre en parallèle sur deux journées), puis la présentation rapide des papiers retenus, en parallèle avec d'autres conférences invitées, pour enfin se terminer par une présentation (très rapide) de travaux en cours.

La conférence a rassemblé un peu plus de 300 personnes, soit sensiblement moins que les éditions précédentes, avec très peu d'européens ou même de non-américains.

1 Tutoriels

Deux journées de tutoriels étaient programmés en parallèle :

Lundi

- Intrusion Detection and Prevention Systems (Marcus Ranum)
- Logging & Security, building an enterprise logging infrastructure (Tina Bird)
- WiFi Security (William Arbaugh)
- dDoS attacks and defense (Jelena Mirkovic & Peter Reiher)

Mardi

- Building Honeypots for intrusion detection (Marcus Ranum)
- Hacking and securing web applications (David Rhoades)
- Network Security Protocols, theory and current standards (Radia Perlman)
- Using FreeBSD's advanced security features (Mike DeGraw-Bertsch)

Les tutoriels reflètent l'intérêt que porte la communauté à certains sujets. Outre les incontournables de ce type de conférence, comme IDS et Honeypots, on peut noter toujours l'intérêt envers la sécurité des applications Web, du WiFi, et des journaux. Bien que le titre soit assez prometteur, ce tutoriel ne couvre cependant que l'état de l'art des divers journaux produits par les produits les plus courants.

Le tutoriel dDoS a permis de faire le point sur les différentes techniques proposées par les produits du marché ou de recherche, en faisant ressortir le compromis contrainte de déploiement/efficacité.

Mike DeGraw-Bertsch a fait le tour de la sécurité dans FreeBSD, c'est à dire les éléments classiques sous Unix, et des spécificités comme les jail ou TrustedBSD (Mandatory Access Control).

2 Sujets de recherche

Les nouveaux projets de recherche sont divisés en plusieurs thèmes. Cette partie résume l'immense majorité d'entre eux. Le lecteur se reportera aux papiers originels pour de plus amples informations, en sachant que certains sont des sujets de recherche n'ayant pas forcément d'application immédiate possible dans l'industrie.

2.1 Attaques

Remote timing attacks are practical [1]

Ce papier montre que ce type d'attaque, afin de récupérer les clefs privées en mesurant les temps de réponse des serveurs, sont réalisables. D'autant plus que des études montrent que des attaquants se situent sur des noeuds internet.

802.11 Denial of service attacks [2]

Cette étude présente de manière pratique des dénis de service sur les réseaux sans fil, en utilisant d'une part les réservations de canaux pour un temps donné, et d'autre part de forcer la désauthentification des clients continuellement. L'étude a par ailleurs montré que toutes les puces 802.11 du marché dérivait du même masque, réalisé au départ par Choice Microsystems, et que certains défauts anciens se trouvent donc partout.

Denial of Service via Algorithmic Complexity Attacks [3]

Cette étude montre comment utiliser les algorithmes, notamment de hash, pour réaliser à peu de frais un déni de service, en rendant le temps d'exécution exponentiel. Les hash "universels" comme MD5 ou le système inclus dans Perl, sont également vulnérables. Ils sont choisis à cause de leur comportement dans le meilleur cas, en oubliant le pire cas (*worst case*), c'est à dire dans les situations où des collisions peuvent se réaliser. Ceci peut par ailleurs impacter potentiellement beaucoup d'applications embarquées, comme dans les routeurs ou les firewalls.

2.2 Durcissement (Hardening)

Il s'agit ici de protéger les logiciels au niveau de leur code, une protection contre les débordements de buffer majoritairement.

Pointguard : Protecting Pointers from Buffer Overflows Vulnerabilities [4]

Pointguard permet de chiffrer les pointeurs lorsqu'ils sont en mémoire, pour ne les déchiffrer momentanément que lorsqu'ils sont chargés dans un registre du CPU. Cet outil s'insère dans un compilateur, ici GCC, et implique toutefois une performance variable, de 0 à 20%.

Address Obfuscation : An efficient approach to combat a broad range of memory error exploits [5]

Address Obfuscation est une autre technique, qui consiste à rendre aléatoire les localisations du code et des données en mémoire. Ainsi, une attaque réussie de débordement de buffer ou autre, qui doit en général utiliser ces informations, ne sera pas réutilisable sur un autre système. Ceci est relatif à ce que peut faire PaX [6]

High Coverage Detection of Input-Related Security Faults [7]

Le dernier travail présenté est plus de la détection. Il est possible de faire une analogie avec le mode Tainted de Perl, qui vérifie l'état des variables. Leur approche a permis de détecter plusieurs vulnérabilités dans des programmes (dans des versions vulnérables), comme OpenSSH.

Preventing Privilege Escalation [8]

Improving Host Security with System Call Policies [9]

Ces deux présentations de Niels Provos sur la séparation des privilèges et sur le filtrage des appels systèmes (systrace) ne sont pas réellement nouvelles, mais n'avaient pas été présentées à Usenix.

Le premier consiste à séparer dans un programme le code qui doit s'exécuter avec des privilèges, et le reste. Ainsi, les erreurs de programmation dans la partie s'exécutant sans aucun privilège (et en cage) auront beaucoup moins d'impact. Le second travail consiste à, de manière analogique avec un *firewall*, effectuer un contrôle par une politique sur les appels système.

Dynamic Detection and Prevention of Race Conditions in File Accesses

Les race conditions (TOCTTOU : Time Of Check To Time Of Use) sont un problème récurrent. Cette méthode propose un examen des opérations sur les systèmes de fichiers (ouverture, unlink, etc.), qui couplé à une politique prédéfinie permet de détecter et/ou d'empêcher la plupart de ces attaques. Ceci induit naturellement un problème de performance, notamment sur la fonction open().

2.3 Détection

Storage-Based Intrusion Detection : Watching Storage Activity for Suspicious Behavior [10]

Le premier système présenté est un système de détection d'intrusion sur les systèmes de stockage (Storage-based intrusion detection). Il s'agit de 100 lignes de code ajoutées au serveur NFS, qui permettent de détecter les accès (journaux, fichiers importants), le comportement, et les données suspectes. L'expérience a montré la détection de nombreux rootkits ou vers, avec une perte de 0,1 à 1% de performances sur l'utilisation du serveur.

Detecting Malicious Java Code Using Virtual Machine auditing [11]

Il s'agit de produire un flux d'événements, au niveau de la JVM, incluant les appels système, les événements de classe (construction, destruction, ...), les événements JNI et l'interaction des différents threads. Enfin, plusieurs scénarios de comportements sus-

pects sont écrits, comme le scan de ports, le transfert de données sensibles, etc., qui sont détectés grâce à ce flux d'événements.

Static Analysis of Executables to Detect Malicious Patterns [12]

Il s'agit là encore de détecter les comportements suspects des programmes, en analysant la structure sémantique du code machine. En réalisant un graphe de flux, et en détectant les instructions inutiles servant à modifier les signatures et le programme (NOP, modification de registres morts, PUSH EAX POP EAX, etc.), il est possible de détecter des comportements suspects à l'exécution, comme les sauts inutiles, etc. De plus, cette méthode est dépendante du type de processeur et non du langage en amont ou du système.

2.4 Cryptographie (Applied Crypto)

SSL Splitting : Securely Serving Data from Untrusted Caches

Cette technique permet de libérer les serveurs HTTPS d'une partie de leur charge, au moyen de relais cache comme cela est fort utilisé pour HTTP. Le mode de chiffrement bout en bout n'autorisant pas cela, il s'agit pour le relais de simuler la connexion SSL, et de construire le protocole en fonction des informations d'authentification du serveur et des données du cache ou du serveur. Aucune modification n'est nécessaire sur le client.

A New Two-Server Approach for Authentication with Short Secrets [13]

Il s'agit d'une proposition de méthode pour l'authentification via des mots de passe faibles, comme les codes PIN des téléphones portables. L'utilisation de plusieurs serveurs et la façon dont les mots de passes sont partagés permet de durcir le système en cas de compromission du serveur d'authentification.

Domain-Based Administration of Identity-Based Cryptosystems for Secure E-Mail and IPSec [14]

Cette méthode permet de réduire considérablement les actions utilisateurs pour les courriers électroniques et le trafic IP chiffrés, en utilisant IBC (Identity-Based Cryptography) conjointement au DNS.

2.5 Divers (The Road Less Traveled)

Scrash : A System for Generating Secure Crash Information [15]

Scrash répond à un besoin récurrent de sécurité : comment envoyer des dumps (fichier core par exemple) lorsqu'un programme plante à son fournisseur, sans que celui-ci ne contienne d'information sensible. La méthode employée est d'identifier au moyen d'une

fonction spéciale dans le code, ces données, et de nettoyer ensuite le dump. Cependant, lorsque ce besoin existe, c'est que le code n'est pas maîtrisé. Lorsque l'on a des doutes sur le fournisseur, peut-on être sûr que ce type de méthode serait effectivement correctement employée si on le dit ?

Implementing a Virus Throttle Model [16]

Ce système permet de combattre les virus qui se propagent rapidement par le réseau, en limitant dynamiquement le nombre de connexions effectuées. Le modèle a pu être testé avec les protocoles TCP, UDP, SMTP et Exchange.

3 BoFs

Plusieurs BoF d'intérêts divers ont été organisées : sur la journalisation, SELinux, Sun (probablement la plus intéressante), TCPA, ...

Sun a présenté les nouveautés sécurité de Solaris, à savoir :

- Packages granulaires, plus facile à minimiser
- Un remaniement des permissions par défaut
- TCP Wrappers dans inetd
- SunScreen 3.2 intégré
- Mots de passe avec MD5, Blowfish
- SSH basé sur OpenSSH
- Kerberos v6
- IPSec DES/3DES par défaut
- /dev/random et /dev/urandom
- L'isolation : il s'agit d'une fonctionnalité intéressante nommée *zones*[17], à la manière des *jails* de BSD.

4 Divers

4.1 Keynote

Est-il réellement besoin d'en parler ? La keynote a été présentée par *Black Unicorn* (A.S. von Bernhardt), il s'agissait de réflexions sur les notions d'identité et de réputation.

4.2 NGSCB

Plusieurs présentations et une table ronde ont été organisées sur NGSCB (Palladium). A ce titre, deux avis contradictoires sont intéressants :

William A. Arbaugh de Universit du Maryland, pense que NGSCB a été mal présenté et donc mal compris. Il s'agit d'un contrôle d'accès obligatoire (MAC : *Mandatory Access Control*, dans lequel la gestion des droits numériques (DRM : *Digital Rights Management* est un sous-ensemble. Un système MAC applique une politique d'accès, la question est de savoir quelle est-elle et surtout qui l'applique ; car il s'agit d'une bonne chose si la politique de sécurité de l'entreprise est appliquée. Par ailleurs selon lui, cela ne répond pas aux problèmes de virus et de vers.

Douglas Barnes [18], fondateur de C2Net, qui est redevenu étudiant en droit après la revente de sa société, pense lui que NGSCB sera l'amplificateur de la puissance déjà acquise sur le marché, et que cela est conçu pour aboutir une propriété quasi-absolue. Le choix d'activation ou non de celui-ci est illusoire, car les logiciels de type Office le requièreront rapidement pour simplement fonctionner. Ainsi, l'utilisateur n'a que l'illusion de la propriété de son ordinateur et n'a plus aucun droit sur celui-ci. Les abus étant fortement prévisibles et inévitables, seule la force publique peut réguler au travers de la loi.

4.3 Tags RFID

Il a également été beaucoup question des tags RFID [19]. Il s'agit de micro-éléments qui s'insèrent partout, dans les voitures, ordinateurs portables, billets de banques, vêtements, pneumatiques, colliers d'animaux, ...

Ayant pour origine les systèmes antivols, la particularité est de ne pas avoir besoin de source d'énergie. En effet, le signal reçu est utilisé pour produire l'énergie, le transformer et en réémettre un autre.

5 Travaux en cours

Plusieurs travaux en cours ont été présentés, parmi lesquels certains n'étaient d'ailleurs pas très nouveaux, comme honeyd. Il s'agit de sessions courtes de strictement 5 minutes, encadrées par un gong. Morceaux choisis.

Analysis of an Electronic Voting System (Adam Stubblefield) [20]

Les machines à voter encore à l'honneur, en montrant les faiblesses cryptographiques du système, car le code source des bornes utilisées pour les élections aux Etats-Unis est téléchargeable.

Using Link Cuts to Attack Internet Routing (Steve Bellovin) [21]

Steve Bellovin a présenté les attaques par coupure de liens. Il s'agit, sans s'attaquer aux routeurs, de couper (physiquement ou virtuellement) les liens pour, en fonction de la table BGP, obliger à passer par certains noeuds contrôlés par l'attaquant. Les attaques classiques sur le routage peuvent être contrées par de la sécurité de routage, comme l'authentification MD5 ou S-BGP, ce dernier n'étant toujours pas d'actualité. Une contre-mesure proposée est la modification des métriques grâce à des IDS.

Stream : a transparent encryption system (Simson L. Garfinkel)

Il s'agit d'un système de chiffrement transparent pour l'utilisateur. Transparent car le système de gestion des clefs s'attache aux courriers électroniques, par une analogie avec HTTPS, à travers un relais SMTP ou POP.

Wireless LAN Location-Sensing for Security Applications (Algis Rudys) [22]

La détection à 3,5 mètres de précision dans un WLAN au travers de deux nouvelles méthodes (*histo* et *diff*), même lorsque les bornes sont conçues pour empêcher la détection.

Trends in Denial of Service Attacks (Jose Nazario)

Les attaques en déni de service utilisent plus de nos jours UDP que TCP. Si la durée moyenne de celles-ci est constante, il n'en est rien du nombre de paquets par seconde ou de la bande passante.

Denial of Service through Regular Expressions (Scott Crosby) [23]

Les fort utiles expressions régulières peuvent être victime de dénis de services selon leur complexité, leur faisant prendre un temps exponentiel.

Remerciements

- l'OSSIR [24]
- Hervé Schauer
- Denis Ducamp
- Scott A Crosby

Références

- [1] http://www.usenix.org/events/sec03/tech/brumley/brumley_html/index.html
- [2] <http://ramp.ucsd.edu/~bellardo/pubs/usenix-sec03-80211dos-color.pdf>
- [3] http://www.cs.rice.edu/~scrosby/hash/CrosbyWallach_UsenixSec2003.pdf

- [4] http://immunix.com/~crispin/pointguard_usenix_security2003.pdf
- [5] http://seclab.cs.sunysb.edu/addr_obfs/docs/ao.pdf
- [6] <http://www.grsecurity.net/>
- [7] <http://www.eecs.umich.edu/~larsone/security.pdf>
- [8] <http://www.citi.umich.edu/u/provos/papers/privsep.pdf>
- [9] <http://www.citi.umich.edu/u/provos/papers/systrace.pdf>
- [10] <http://www.pdl.cmu.edu/PDL-FTP/Secure/usenix03.pdf>
- [11] <http://www.cs.ucsb.edu/~ckrintz/papers/usenix03.pdf.gz>
- [12] <http://www.cs.wisc.edu/wisa/papers/safeTR1467/cj03.pdf>
- [13] <http://developer.rsasecurity.com/labs/nightingale/files/nightingale-paper.pdf>
- [14] <http://www2.parc.com/csl/members/gdurfee/ibeipsec.pdf>
- [15] <http://www.cs.berkeley.edu/~nks/crash/crash-usenix.pdf>
- [16] <http://www.hpl.hp.com/techreports/2003/HPL-2003-103.pdf>
- [17] <http://www.theregister.co.uk/content/53/30179.html>
- [18] <http://www.salguod.com/>
- [19] <http://www.aimglobal.org/technologies/rfid/>
- [20] <http://avirubin.com/vote>
- [21] <http://www.research.att.com/~%7Esmb/papers/reroute.pdf>
- [22] <http://www.cs.rice.edu/~arudys/papers/wise2003.pdf>
- [23] <http://www.cs.rice.edu/~scrosby/hash/slides/USENIX-RegexpWIP.2.ppt>
- [24] <http://www.ossir.org>