Compte-rendu du SSTIC 2007

rédigé le 4 juin 2007 par Bruno Kerouanton (http://bruno.kerouanton.net)

RCJU / SDI - rue de la Justice 2 - 2800 Delémont, Suisse

Résumé : Cette cinquième édition du SSTIC (Symposium sur la Sécurité des Technologies de l'Information et des Communications) qui se déroulait comme chaque année à Rennes, était d'un niveau sensiblement plus élevé que celui des années précédentes, avec notamment un fort accroissement des présentations concernant les aspects « bas niveau » et mettant en jeu les aspects matériels ou liés aux processeurs et noyaux.

Globalement, ces 3 jours ont permis de démontrer que l'activité recherche en SSI est toujours en ébullition, et que malgré les efforts déployés pour protéger ses environnements, les attaques en tous genre sont toujours d'actualité et il convient de rester extrêmement prudent lors de la mise en oeuvre ou de l'utilisation de nouvelles technologies et de systèmes d'information.

Et pour la première fois, les meilleures conférences techniques seront publiées dans un numéro special de la revue de recherche Journal of Computer Virology, en anglais. Comme quoi le niveau technique de SSTIC a explosé ![2]

Table des matières

1Documents disponibles librement ici :

http://bruno.kerouanton.net/papers/sstic2007-mai07-bk-synthese.pdf

http://bruno.kerouanton.net/papers/sstic2007-mai07-bk-synthese-rumps.pdf

2 me souffle son éditeur en chef !

Mercredi 30 mai 2007 - matin

Avant d'aborder le vif du sujet, il est à noter que cette année l'ensemble des places disponibles pour assister à cette édition du SSTIC a été vendu en moins de 3 jours, ce qui prouve que le SSTIC est devenu un événement incontournable pour les experts du monde de la sécurité des systèmes d'information en France.

RSSI au sein du Conseil Constitutionnel

La session d'ouverture évoquait les risques liés aux systèmes d'information pour un organisme tel que le Conseil Constitutionnel, organe régulateur de l'Etat en matière législative.

On y apprenait par exemple que la fonction sécurité y est représentée par une seule personne à court de moyens ; qu'il existe une version authentifiée (par signature électronique) du Journal Officiel électronique mais que personne ne s'en sert, la version non authentifiée présente via le site legifrance.gouv.fr étant plébiscitée.

L'orateur a émis quelques questions ouvertes concernant le risque informatique encouru par le Conseil Constitutionnel lors des élections présidentielles, l'adoption d'ELAO (Elaboration de la Loi Assistée par Ordinateur), du vote électronique, et de la justice par visio-conférence dans les Dom-Tom.

Une présentation en fait un peu terne, présentant quelques risques potentiels et éléments liés aux soucis du RSSI en poste mais sans vraiment d'éléments percutants. D'autant plus que la session de questions-réponses finale a éte stérile, laissant penser que l'orateur n'était pas au courant de l'activité réelle du responsable sécurité.

Signature électronique : date et lieu

La seconde présentation portait un titre pouvant évoquer un roman de science-fiction pour le néophyte : « vers un marquage spatio-temporel des documents électroniques ». En fait, il s'agissait plus simplement de l'élaboration d'une méthode permettant l'horodatage et l'authentification du lieu d'émission de documents électroniques.

Il existe en effet d'ores et déjà des méthodes et protocoles d'horodatage (« *timestamping* ») permettant de garantir le moment précis où est signé électroniquement un document électronique, mais on ne garantit pas le lieu de la signature. Les chercheurs présentant leurs travaux en la matière ont imaginé un « protocole topographieur » permettant également de garantir le lieu de la signature, en plus de la date.

Si l'intérêt présenté par cette démarche semble évidente, il reste malgré tout à savoir si cela sera mis en application, ce travail étant purement académique. De plus, la signature électronique telle qu'on la connaît n'inclut que l'horodatage et pourtant semble suffire à tous, il est donc probable que l'ajout de la dimension spatiale soit difficile à justifier, du moins dans un premier temps, car cela reviendrait à revoir l'ensemble des mécanismes actuellement en place.

Comment passer administrateur du domaine en 30 secondes...

Pour nous mettre en appétit avant le déjeuner, Aurélien Bordes nous a fait une très belle démonstration des (très mauvaises) méthodes de stockage des mots de passe et des condensats par Windows.

Après un exposé théorique nous rappelant les bases de la gestion des mots de passe et de l'authentification par Windows (LM, NTLM, NTLMv2), Aurélien a abordé leurs faiblesses et nous avons pu suivre en vidéo quelques attaques sympathiques, notamment l'ouverture de sessions « Administrateur de Domaine » en quelques secondes grâce à la mauvaise gestion par Windows des données secrètes.

Voici en résumé le principe : D'une part, lorsque deux comptes sont ouverts successivement sur un poste Windows, celui-ci n'efface pas le condensat associé. D'autre part Windows n'utilise pas le mot de passe mais le condensat associé pour effectuer les transactions d'authentification réseau. On en déduit donc qu'il est facile de récupérer le condensat de la session administrateur, puis de s'en servir pour s'authentifier en tant que tel sur le contrôleur de domaine et en prendre le contrôle des ressources. Le tout sans connaître le mot de passe.

Et l'on n'ose même pas évoquer la gestion lamentable des mots de passe sous leur forme « LM », qui permet leur cassage quasi-instantané (si ils n'ont pas d'accents... sauf pour HSC !), d'autant plus que ces derniers sont automatiquement et invariablement calculés même si cela n'est pas nécessaire...

Une belle et didactique présentation qui risque d'inquiéter plus d'un administrateur de parc Windows !

Mercredi 30 mai 2007 - après-midi

Rétroconception et cartes à puce

Le début de cette première après-midi était consacrée à la sécurité matérielle. Christophe Clavier, chercheur chez Gemalto nous a tout d'abord présenté les différentes techniques d'attaques permettant de récupérer les clefs privées stockées secrètement sur des puces cryptographiques RSA, avant de se focaliser sur l'attaque DPA (« *Differential Power Analysis* ») et la relative aisance qu'un attaquant possède pour visualiser chacun des bits de la clef RSA privée.

En effet, l'algorithme de calcul utilise l'exponentiation modulaire, elle-même basée sur l'algorithme *square-and-multiply*, ce dernier ayant une forte tendance à consommer (ou pas) du courant suivant que le bit présenté est à 0 ou à 1... La mesure (avec un équipement adéquat que le quidam moyen ne possède certainement pas) est alors possible.

Après cette introduction sans grande surprise, la méthode étant déjà connue depuis quelques années, Christophe Clavier a ensuite évoqué la généralisation du principe pour découvrir des clefs et des protocoles cryptogaphiques (presque) inconnus, tels que A3/A8 utilisé en teléphonie GSM et dont l'implémentation propriétaire est laissée libre à chaque opérateur.

La méthode d'attaque SCARE (« *Side Channel Analysis for Reverse Engineering* ») a été mise au point dans ce but, et semble prometteuse, cependant il m'a semblé que Christophe Clavier reste focalisé sur des attaques très précises et que tous ces travaux ne tiennent pas compte des avancées récentes en anti-tampering mises en place dans la plupart des nouvelles puces de chiffrement, capables par exemple de résister aux injections de fautes ou aux attaques DPA.

Beau travail théorique mais probablement très difficile à utiliser en conditions réelles, vu le matériel et le temps nécessaire : retrouver ladite clef suppose de renouveler l'attaque plusieurs milliers de fois et de comparer les résultats individuellement... d'ailleurs la question piège a été posée dans la salle à l'issue de la présentation : Comment fait-on lorsqu'il y a un code PIN ou un verrouillage au bout de 3 essais ? L'absence de réponse du chercheur était éloquente et en a fait sourire certains.

Enfin un processeur sécurisé !

Cryptopage est un processeur ayant pour caractéristique majeure la sécurité et élaborée par des thésards de l'ENST pour le compte de la Délégation Générale de l'Armement.

En gros, ce processeur considère tout ce qui est externe (bus de données et d'adresses, mémoire, stockage etc.) comme pouvant être sous contrôle d'un attaquant. Il va alors permuter et chiffrer tout ce qui y transite, en obfuscant les allocations mémoire, empêchant la plupart des attaques connues (rejeu, débordement de tampon, lecture mémoire, injection etc.) par le biais d'une infrastructure appelée HIDE (« *Hardware support for leakage-Immune Dynamic Execution* »).

Sur le papier c'est intéressant, notamment pour l'exécution de code protégé ou sensible de manière distribuée sur des systèmes non contrôlés (peer-to-peer, grid computing etc.). Malheureusement la quasi-totalité de nos ordinateurs utilisent des processeurs non sécurisés et on voit mal comment les principaux constructeurs (armée exceptée) se mettraient à intégrer un processeur CryptoPage sur les cartes mères.

Une solution envisagée serait de le proposer sous forme de machine virtuelle, ce qui serait certainement très intéressant. Comme ces travaux ont été réalisés pour la DGA il est également possible que ce modèle de processeur soit utilisé « en interne », ce qui du coup justifie son développement.

Englué dans un pot de miel

Les pots de miel permettent d'attirer les attaques pour observer les techniques employées. Leur principal inconvénient est que le pirate va très souvent « butiner » de machine en machine, et que le pot de miel ne représente que l'une d'entre-elles, rendant l'observation imparfaite et parcellaire, donc souvent peu intéressante.

L'idée implémentée par les 5 chercheurs de l'équipe LAAS du CNRS consiste donc à mettre en place des mécanismes de redirection dynamique leurrant l'attaquant souhaitant « butiner » de machine à machine, lui faisant croire qu'il se promène sur le réseau alors qu'il est toujours sur un pot de miel. Cela est réalisé via des hooks NetFilter et le module conntrack, (choix qui n'a d'ailleurs pas semblé très judicieux par certains experts, d'après les questions qui ont suivi la présentation). Une sorte de manière de l'engluer, finalement.

L'équipe admet néanmoins dans son papier que le principe possède un certain nombre de limitations et que le pirate risque assez vite de se rendre compte de la supercherie, mais le mécanisme étant opérationnel depuis quelques mois ils ont pu malgré tout collecter quelques exemples d'attaques.

Encore un travail académique potentiellement intéressant mais méritant des efforts importants pour être réellement utilisable.

Voyage dans l'immensité d'IPv6

La journée se clôturait par une présentation intéressante d'un consultant HSC. Les réseaux IPv6 ne sont pas encore vraiment répandus (il paraît qu'un jour cela viendra... il paraît... un jour !), mais il est bien connu que l'une des caractéristiques majeures de ce réseau est l'absence de classes d'adresses, et par conséquent un nombre absolument faramineux d'adresses IP au sein du réseau.

Le problème exposé par Nicolas Collignon est donc de parvenir à réaliser une découverte de réseau Ipv6 ainsi qu'une cartographie associée, sachant qu'un « *scan* » utilisant les méthodes classiques prendrait selon lui un demi milliard d'années, et que certaines méthodes ou outils existant en IPv4 ne sont plus d'actualité en Ipv6.

Nicolas propose donc différentes techniques basées sur les mécanismes d'auto-découverte, d'interrogation DNS et de scans multicast, le tout concentré au sein de son outil Sherlock. Ce dernier fonctionne en mode distribué et est basé sur trois catégories de robots spécialisés ainsi que sur un ensemble de sondes, utilisant une bonne dizaine de techniques distinctes pour topographier le réseau et corrélant le tout.

Un outil prometteur mais qui nécessite encore quelques optimisations car étant assez récent.

Jeudi 31 mai 2007 - matin

Le Wi-Fi c'est dangereux !

Beaucoup de mes amis et collègues attendaient avec impatience cette présentation sur le fuzzing des drivers Wi-Fi. Ils avaient raison !

En synthèse, une très belle démonstration des attaques en « boîte noire » sur les pilotes de cartes Wi-Fi (plus précisément du driver de la puce Broadcom présente sur une grande majorité de cartes), afin d'y découvrir des failles. L'intérêt de la méthode est grandiose : d'une part les drivers étant chargés avec des droits privilégiés par le système, une simple compromission permet à l'attaquant d'exécuter du code arbitraire en mode noyau (ring 0), mais d'autre part, et c'est le plus alléchant, cela peut se faire à distance via radio, en attaquant directement la couche radio du pilote...

Verdict : le simple fait d'activer le Wi-Fi sur son ordinateur équipé d'une puce Broadcom le rend vulnérable à une exécution de code arbitraire et l'attaquant peut ainsi en prendre le contrôle quasiment immédiatement, comme cela a été démontré en direct (la démonstration étant basée sur une vulnérabilité du pilote Madwifi permettant de lancer un shellcode).

Angoissant, n'est-ce-pas ? (et il ne faut pas compter sur WPA pour s'en sortir puisque l'attaque concerne les couches « basses », donc avant les aspects chiffrement...)

Attaques du noyau Linux

Stéphane Duverger présente les méthodes avancées pour prendre le contrôle de l'espace noyau sous Linux 2.6.

Il s'agit d'améliorations et d'optimisation des méthodes classiques généralement utilisées pour prendre le contrôle en mode applicatif (« userland »), mais comme il s'agit de l'espace noyau c'est

sensiblement plus complexe et il faut ruser. En contrepartie une exploitation de ce type permet d'obtenir un vrai accès total au système, ce qui est d'autant plus intéressant.

Après un bref rappel sur les mécanismes de gestion des tâches et de la mémoire par le noyau, Stéphane nous explique comment gérer les appels système une fois en espace kernel, lancer un shellcode, et modifier l'espace d'adressage. Son papier livré dans les actes du SSTIC est très détaillé et clair.

Suivra un exemple (complémentaire de la présentation précédente) d'infection du driver Wifi Broadcom sous NDISwrapper.

Rootkits invisibles

Cela semblera un pléonasme pour certains, qui ont appris qu'un rootkit était par définition invisible. Mais d'autres le savent bien, sous certaines conditions la plupart des rootkits peuvent se révèler au grand jour, ou au moins laisser des signes de présence.

Eric Lacombe, Fred Raynal et Vincent Nicomette ont planché sur le sujet et traitent ici de la capacité à réaliser un rootkit totalement furtif sous Linux. La présentation décrit en détail l'architecture et les méthodes d'installation des rootkits standard, ainsi que leurs modes opératoires notamment lorsqu'ils doivent se rendre « visibles » quand l'attaquant les active ou communique avec eux.

Ensuite, ils présentent leur propre rootkit furtif, utilisant différentes techniques évoluées telles que le parasitage des threads noyau ou le fait d'attaquer directement la mémoire via /dev/kmem pour se dissimuler au mieux.

Encore une fois, démonstration intéressante d'une preuve de concept qui peut en inquiéter plus d'un.

Analyse statique de codes

Lorsqu'il s'agit de déterminer si un programme possède des failles de sécurité, il existe plusieurs approches. Le fuzzing est à la mode mais est plutôt aléatoire et grossier.

Xavier Allamigeon et Charles Hymans nous ont présenté une méthode quasi-formelle d'analyse des programmes afin d'en déterminer les failles. Cela s'effectue à l'aide de désassemblage du code puis de sa modélisation mathématique afin de représenter au final des aires shématisant l'exécution stable ou non. L'utilisation du langage NewSpeak mis au point pour l'occasion permet la formalisation et le passage du monde réel au monde abstrait.

Tant la présentation que le papier fourni dans les actes sont remarquables, mais nécessitent un certain éveil de son esprit pour une bonne compréhension. Il est certain que ce type de travaux de recherches va permettre d'améliorer grandement la sécurité des développements. Tant la présentation que le papier ont d'ores et déjà été primés par le comité de programme du SSTIC, et Eric Filiol a félicité les auteurs au nom de la communauté scientifique pour les avancées qui découleront de leurs travaux.

Metasm

L'assembleur-désassembleur-dynamique-a-tout-faire, tel pourrait être le nom de l'outil Metasm écrit en Ruby (tout comme Metasploit).

Les langages évolués de type Python ou Ruby sont à la mode pour écrire des outils de génération

ou de manipulation de données dynamiques, et Metasm s'inscrit donc dans la lignée de Scapy ou de Wifitap, par exemple. Il permet notamment en temps réel le cycle de désassemblage, de modification de code puis de réassemblage avant exécution afin de permettre des changements de comportement de l'application testée.

Cela peut notamment servir à générer à la volée des shellcodes polymorphiques conjointement avec Metasploit, à faire du fuzzing sous certaines conditions, et bien d'autres choses.

Ce que l'on peut retenir en tout cas, est qu'il devient de plus en plus nécessaire de maîtriser les langages Python et Ruby, vu la qualité des outils tournant avec.

Jeudi 31 mai 2007 – après-midi

Les secrets des disques durs

Laurent Dupuy a ensuite présenté les subtilités non (ou peu) documentées des disques durs ATA et SATA utilisées par tout le monde.

Il existe en effet un certain nombre de commandes interprétables par le processeur intégré sur le disque (rappelons que celui-ci comporte son propre microcode), permettant l'effacement automatique, mais également la reconfiguration des caractéristiques du disque afin par exemple de le rendre compatible avec un modèle de moindre capacité, ou afin de le brider vis-à-vis de l'utilisateur.

Cela est également délicat en cas d'enquête forensics, car une personne partioculièrement malveillante et au fait pourrait reconfigurer le disque pour masquer certains fichiers. Un numéro de série est également accessible, tout comme différentes zones réservées (secteurs défectueux, espace protégé,...) et des systèmes de protection par mot de passe... désactivables assez aisément si l'on a accès à la carte électronique.

Conclusion : mieux vaut chiffrer intégralement son disque en utilisant des outils logiciels.

Forensics mémoire

Comme chaque année Nicolas Ruff nous a présenté un sujet intéressant. Cette fois-ci, son discours était un peu moins « risqué » que les années précédentes, mais néamoins clair.

Dans le cadre des enquêtes judiciaires, il est parfois nécessaire de ne pas « couper le courant » afin de récupérer et d'analyser la mémoire du système suspect. En effet, de plus en plus d'attaques se font par injection directe en mémoire sans passer par le disque, et les traces sont donc inexistantes.

Parmi les problèmes les plus importants liés à l'analyse mémoire, on retrouve la lenteur pour dumper son contenu (surtout via réseau), son caractère dynamique et fluctuant rendant l'image non intègre, et bien d'autres soucis.

Diverses solutions existent mais aucune n'est satisfaisante, car nécessitant l'installation d'une carte (impensable dans le cas d'un enquête), le forçage d'un CrashDump (incomplet), l'accès DMA via bus Firewire (ne permet pas de tout récupérer et peut aussi planter complètement le système), etc. Bref, pas simple. Nicolas confirme que le seul véritable moyen de récupérer la mémoire de manière intègre est dans le cas de la machine virtuelle suspendue.

La présentation continue avec deux exemples d'analyse de la mémoire et la présentation d'outils dédiés (onéreux, comme tout ce qui touche aux Forensics, selon Nicolas !), et conclut sur les techniques anti-forensics en mémoire. Intéressant, n'est-ce pas ?

Actualité juridique

La dernière présentation de la journée nous était présentée par la désormais habituée Marie Barel, puisqu'il s'agit de sa quatrième intervention au SSTIC. Cette fois-ci, elle nous a présenté les aspects du chiffrement et la notion documents privés en entreprise.

En synthèse, la première partie de son exposé traitait de la libéralisation progressive de l'usage des outils de chiffrement en France, et notamment les aspects exportation de produits de chiffrement français à l'étranger, systématiquement soumis à déclaration voire à autorisation préalable auprès de la DCSSI, ce que beaucoup de personnes ne savent pas (mais nul n'est censé ignorer la loi, nous rappelle t'elle !).

La seconde partie de son exposé traitait les problèmes liés à l'utilisation d'outils de chiffrement par les salariés au sein de l'entreprise, et notamment des cas particuliers d'employés chiffrant par le biais d'outils apportés certaines données « personnelles », les mettant alors hors de portée de l'entreprise. Le seul moyen pour l'employeur de s'en sortir est d'empêcher cela par le biais de la charte informatique.

Autre élément intéressant fourni par Marie Barel : une suite donnée au fameux « arrêt Nikon », la jurisprudence récente permettant désormais de présumer que tout document présent au sein de l'entreprise est à caractère professionnel sauf mention contraire, ce qui va permettre à l'employeur d'exercer plus facilement une certaine cybersurveillance.

Et en guise de conclusion, Marie Barel nous a annnoncé sa prochaine arrivée chez Silicomp/AQL/Orange.

Rump Sessions

Au nombre de 25 cette année, soit 4 minutes par rump session, le record a été battu. Compte-tenu de ce nombre, j'en ferai le détail dans un document annexe à ce compte-rendu. Comme d'habitude de belles surprises, des rump passionnantes, des révélations, et des présentations amusantes !

Suivait le Social Event, (longue) soirée permettant de faire connaissance et de discuter en détail de tout cela, pour ceux et celles qui ont eu la chance de s'inscrire parmi les premiers.

Vendredi 1er juin 2007 – matin

Les CESTI

La première présentation de la journée traitait en détail (pour ceux et celles qui ont été raisonnables et ne se sont pas couchés trop tard) des aspects de la certification logicielle par un centre agréé CESTI. L'orateur, Christian Damour, est un spécialiste en la matière car il travaille au sein d'une société étant CESTI.

Après avoir brossé les différentes méthodes et référentiels de certification (Critères Communs, etc.), Christian nous met en garde sur les limites et subtilités, notamment la mauvaise (ou volontairement restreinte) définition du périmètre certifié, les évolutions dans le temps, les différentes « qualités » d'évaluation en fonction du pays et de l'organisme, etc.

Très intéressant pour les personnes n'étant pas au courant de ces aspects, cela relativisant fortement les trop fréquents discours commerciaux privilégiant ces fameuses certifications sans en évoquer les détails.

VoIP

Suivait une présentation (à laquelle je n'ai malheureusement pas pu suivre le contenu) sur les trop nombreux risques liés au déploiement de voix sur réseaux IP et notamment des failles relatives au protocole SIP.

Nicolas Dubée présentait ensuite une solution de chiffrement des conversation IP basée sur du matériel afin de résoudre le souci.

Je vous recommande le compte-rendu de Cédric Blancher présent sur son blog (http://sid.rstack.org/blog) pour plus d'informations.

Formats Office XML

Philippe Lagadec nous a présenté les subtilités des nouveaux formats de documents bureautiques, basés sur XML puis zippés (essayez de renommer un fichier Office2007 ou Openoffice en .zip, vous verrez bien !).

Ces formats sont, grâce au sentiment de clarté qu'ils dégagent (c'est du XML après tout), susceptibles de limiter la dissémination d'information sensibles, mais ils ne résolvent pas tout. En effet, dans certains cas ceux-ci embarquent également des fichiers binaires propriétaires (objets OLE, images etc.) que l'on trouvait dans les anciennes versions, et donc difficilement auditables.

La suite OpenOffice est également capable d'exécuter des macros écrites dans différents langages tels que le Python,

De plus certaines caractéristiques notamment des formats XML et de compression permettraient d'insérer des codes malveillants. Néanmoins il semble que le format XML ne soit pas une si mauvaise idée que cela au final car le filtrage et le nettoyage de leur contenu devient plus facile.

Une bonne présentation à mettre en parallèle avec la Rump Session d'Eric Filiol donnée lors du SSTIC 2006 et traitant des risques liées aux macros sous OpenOffice.

Attaques Web de type CSRF

Renauf Feil et Louis Nyffenegger, deux consultants HSC, ont dressé un état de l'art en matière de Cross-Site Scripting, et plus précisément de Cross-Site Request Forgery.

Le principe des attaques CSRF est relativement simple mais pourtant efficace, et joue sur le fait qu'un utilisateur a plusieurs fenêtres de son navigateur ouvertes simultanément sur différents sites.

Si par exemple l'une de ces fenêtres pointe sur une application web avec authentification (interne ou externe à l'entreprise, peu importe), et que l'utilisateur lance une seconde fenêtre et va surfer sur un site malveillant dont l'objectif est précisément de s'attaquer à l'application web de la première fenêtre, le risque est élevé.

En effet, l'utilisateur aura déjà saisi son authentification et son navigateur sera vu comme « autorisé » à aller sur l'application web légitime. La seconde fenêtre peut tout à fait contenir du code malveillant qui effectuera des requêtes vers l'application web avec succès ! Le problème est lié au fait que le navigateur partage un même contexte entre les différentes fenêtres et qu'un site web malveillant peut alors réaliser une action non autorisée dans une application web avec les droits de l'utilisateur et sans son consentement.

L'attaque peut être sophistiquée en utilisant du javascript, si le navigateur ne le filtre pas. Mais

dans tous les cas il existe une limitation importante. Si il est possible de lancer des actions sur le site web légitime, il n'est pas possible d'en récupérer le résultat, et les actions doivent par conséquent être réalisées en aveugle.

L'attaque n'est pas récente, les premiers échos sur le forum Bugtraq datent de 2001. Cependant peu de développeurs d'applications web en tiennent compte lors des développements et il est assez fréquent de pouvoir lancer de telles attaques. Les auteurs de la présentation présentent ensuite un certain nombres de sites vulnérables (dont celui du SSTIC, corrigé depuis !) et nous rassurent cependant en expliquant que les sites de gestion des comptes bancaires en ligne ne semblent néanmoins pas posséder ces soucis.

Le développement des applications « web 2.0 » avec traitement asynchrone des pages, permet également ces attaques CSRF. En effet, la commande « XMLhttpRequest » permet à la page affichée de lancer des requêtes comme précédemment. La théorie voudrait que cela ne soit pas possible car un mécanisme de limitation (« same domain policy ») empêche cela, mais si l'attaquant a le contrôle du serveur DNS, même externe, il peut alors leurrer le navigateur, rendant l'attaque possible.

HSC présente enfin un outil de démonstration de ce type d'attaques, *CSRF-Proxy*, et donne quelques pistes sur les méthodes permettant de limiter ce type d'attaques.

Vendredi 1er juin 2007 – après-midi

Cryptographie et vote électronique

Suite à la session d'ouverture qui avait fait jaser certaines personnes présentes au sujet des machines à voter, il fallait bien qu'une conférence de haut niveau vienne rétablir la vérité. Nous avons eu la chance d'avoir Marc Girault pour cela, cet expert émérite travaillant pour France Telecom R&D nous a très clairement expliqué tout ce que l'on souhaitait savoir sur le vote électronique, les machines à voter et les algorithmes associés.

Tout d'abord, il n'était pas là pour nous vanter les mérites de cette méthode de vote, mais pour nous donner un point de vue objectif, critiquant parfois certains usages et restant parfaitement « les pieds sur terre » lors de son argumentation.

Nous avons appris que le vote électronique est autorisé en France depuis 1961 (et oui !), que les 3 méthodes sont le vote par machine certifiée autonome, par machine certifiée en réseau et par Internet, que cette dernière méthode est d'ores et déjà utilisée notamment en Estonie, que de plus en plus de pays s'y mettent et que cela n'est pas si inquiétant que cela, du moins si l'on respecte les caractéristiques fondamentales du vote électronique.

Celles-ci sont : l'anonymat, l'impossibilité de voter plusieurs fois, l'impossibilité de révoquer son vote, la possibilité de refuser de voter si l'on est sous contrainte ou sous pression, etc. Bref, une série de caractéristiques évidentes lorsqu'il s'agit d'aller mettre son papier dans l'urne après être passé par la case isoloir, mais bien plus complexe « à distance ».

Heureusement les mathématiques et la cryptographie vient à notre secours, et Marc Girault nous présente alors les différents protocoles cryptographiques permettant d'assurer les propriétés requises.

Une très belle présentation, encore une fois, malheureusement celle-ci ne figure pas dans les actes.

Cassage de mots de passe

La présentation finale de cette édition du SSTIC était réalisée par Simon Maréchal, et traitait des possibilités permettant d'optimiser le cassage de mots de passe... à des fins d'audit exclusivement, bien entendu.

Après un rappel des bases et des principaux outils permettant de retrouver des mots de passe en clair à partir de leur condensat (*John the Ripper*, *Bob the Butcher*, *RainbowCrack, Ophcrack, Cain*),

Simon nous explique que les Rainbow Tables peuvent être optimisées en utilisant des méthodes probabibilistes, les chaînes markoviennes, mais il semble qu'aucun outil diffusé n'implémente cela.

Après cette incursion dans la théorie du cassage, nous en venons aux aspects pratiques et notamment la raison pour laquelle le Pentium 4 est le pire processeur dans ce domaine : son pipeline trop long ralentit fortement l'exécution dans certains cas (lorsqu'un registre est modifié ou lu par plusieurs instructions présentes en même temps dans le pipeline, le Pentium doit alors le vider, on perd alors de nombreux et précieux cycles).

Il propose alors des optimisations basées sur la réorganisation des instructions assembleur, voire les fonctions vectorielles MMX et SSE, capables de traiter 128 bits en un seule opération. Cela nous amène naturellement vers la question du « quel processeur choisir ? » pour optimiser au mieux... et Simon nous propose le processeur Cell d'IBM.

Celui-ci est présent dans la PlayStation 3 de Sony. Il est en fait constitué d'un PowerPC 64 relativement lent couplé à 7 coeurs de calcul vectoriel rapides, dont 6 utilisables sous Linux (Sony permettant de transformer sa console PS3 en serveur Linux). Nous avons ensuite droit à une petite démonstration de l'outil sous Linux, un brute-forceur semblant un peu trop rapide, effet démo oblige !

Enfin, la solution préconisée semble néanmoins être l'utilisation de processeurs spécialisés, les FPGA déjà connus dans ce domaine pour le cassage du DES par exemple. Un rapide calcul du rapport rapidité/prix nous oriente également vers les FPGA qui peuvent être parellelisés, mais il est souvent complexe de développer dessus.

En conclusion, une présentation intéressante mais qui laissera bon nombre d'entre nous sur leur fin.

Conclusion

Comme chaque année ma conclusion sera positive. Le SSTIC s'est fait une réputation d'excellence auprès des experts en sécurité des systèmes d'information français, et commence à se faire connaître à l'étranger. Le comité d'organisation a fait un travail excellent et a notamment corrigé certains points négatifs signalés l'année précédente.

Non seulement on y retrouve la plupart des acteurs majeurs du domaine, prétexte pour faire connaissance en allant discuter autour d'un verre lors du Social Event ou des déjeuners à Supélec, mais on y apprend de nombreuses choses et ces trois jours, malgré leur densité, semblent franchement passer à une vitesse trop rapide.

A l'année prochaine !

Liens

Ci-dessous, quelques liens (liste non exhaustive) relatifs au SSTIC 2007 :

Site officiel : http://www.sstic.org

Actes du SSTIC : http://actes.sstic.org

Compte-rendu par Cédric Blancher :

http://sid.rstack.org/blog/index.php/2007/06/04/193-le-sstic-2007-comme-si-vous-y-etiez-ou-pas

Compte-rendu par Bruno Kerouanton :

http://bruno.kerouanton.net/dotclear/index.php/2007/06/04/117-sstic-compte-rendu

Compte-rendu par « Nono » :

http://nonop.blogspot.com/2007/06/sstic-2007.html

Compte-rendu par Guillaume Arcas :

http://yom.retiaire.org/doku.php?id=le_sstic_est-il_toujours_le_sstic

Document changes

v1.0 : version initiale

v1.1 : ajouts

v1.2 : corrections mineures

v1.3 : corrections mineures, précisions, ajout de liens

v1.4 (1 oct 2007) : correction de coquille : c'est Christophe Clavier (Gemalto) qui a traité des aspects attaques matérielles par DPA. Laurent Dupuy (FreeSecurity) nous a présenté les indiscrétions des disques durs. Désolé pour l'inversion.