



HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

Eurosec 2005

S25 – de la vulnérabilité à l'exploit

23 mars 2005

Vulnérabilités logicielles : tendances du moment et solutions possibles

Jean-Baptiste Marchand
<Jean-Baptiste.Marchand@hsc.fr>

- Introduction aux vulnérabilités logicielles
- Cycle de vie des vulnérabilités
 - Exemples
- Enjeux autour des vulnérabilités
- Tendances du moment
 - Vulnérabilités
 - *Exploits*
- Etapes de la gestion des vulnérabilités logicielles
- Conclusion
- Références

- Vulnérabilité logicielle : bogue dans un **produit** ayant un impact sur la sécurité du système d'information
- L'industrie informatique tend à considérer que les bogues sont partie intégrante de tout produit
 - Ce qui s'applique par extension aux vulnérabilités logicielles
 - La prise en compte de la sécurité dans les développements logiciels largement répandus n'est que récente
 - Vulnérabilités systèmes classiques tel que les débordements de buffers restent présentes, au moins dans les systèmes d'exploitation et logiciels serveurs
 - Les applications web introduisent de nouvelles classes de vulnérabilités
 - Côté serveur : applications web ne se protégeant pas correctement des données fournies par des utilisateurs malveillants
 - Côté client : vulnérabilités des navigateurs web et des technologies associées (contrôles ActiveX, applets Java, ...)

- Exemple de cycle de vie des vulnérabilités
 - Un logiciel est publié avec un bogue
 - Le risque peut être identifié en interne ou de façon externe
 - L'éditeur peut être contacté ou non
 - Un correctif est développé et publié
 - Potentiellement, des détails sur la vulnérabilité et une exploitation ont déjà été publiés
 - Potentiellement, des pirates exploitent cette vulnérabilité
 - Tous les systèmes sous **votre** responsabilité sont corrigés
 - **Tous** les systèmes **affectés** sont corrigés
- Référence : Ryan Permeh, eEye
 - <http://www.eeye.com/html/resources/newsletters/versa/VE20050208.html>

- Deux seules étapes systématiques
 - Naissance de la vulnérabilité lors de la sortie du logiciel
 - Disparition de la vulnérabilité lorsque tous les systèmes existants sont corrigés ou que le logiciel disparaît
 - A toutes les chances de ne jamais arriver, de nouveaux systèmes étant installés et non corrigés immédiatement (exemple de CodeRed et d'IIS 5)
- Les autres étapes sont optionnelles et dans n'importe quel ordre
 - Une vulnérabilité peut ne jamais être découverte mais cela ne signifie pas pour autant qu'elle n'existe pas
 - La publication d'un correctif a tendance à accélérer le cycle de vie
 - Tous les chercheurs en sécurité font du *reverse-engineering* sur les correctifs publiés par les éditeurs (ex : éditeurs de d'IDS)
 - Les éditeurs poussent pour retarder la publication de détails sur les vulnérabilités, afin de laisser un délai pour l'application des correctifs

- Vulnérabilités découvertes en interne
 - Vulnérabilités corrigées de façon silencieuse, à l'occasion d'une mise à jour logicielle et restant inconnues des utilisateurs
 - N'aide pas à la diminution du risque, une mise à jour apparemment non nécessaire ayant des chances de ne pas être appliquée
- Éditeur contacté ou non
 - Chercheurs en vulnérabilités responsables contactent l'éditeur lors de la découverte d'une vulnérabilité et attendent la publication d'un correctif avant de révéler des détails sur la vulnérabilité
 - Cas opposé : vulnérabilité maintenue secrète et pouvant être utilisée par une communauté restreinte, qui a un intérêt à la garder privée aussi longtemps que possible (*0day*)
- Systèmes en dehors de votre périmètre mais non corrigés peuvent vous affecter, même si vos propres systèmes sont protégés
 - Cas classique des vers Internet récents, affectant la stabilité de l'Internet

- Vulnérabilité dans l'interface RPC de la LSA des systèmes Windows (MS04-011)
 - Rapportée par eEye à Microsoft : 8 Octobre 2003
 - Correctif MS04-011 publié par Microsoft : 13 Avril 2004 (+ **188 jours**)
 - Publication par eEye des détails techniques sur la vulnérabilité : 13 Avril 2004
 - Exploitation par le vers Sasser : 1er Mai 2004 (+ **17 jours**)
- Vulnérabilité dans le module de décodage du protocole ICQ des sondes de détection d'intrusion (IDS) d'ISS
 - Rapportée par eEye à ISS : 8 Mars 2004
 - Correctif publié par ISS : 18 Mars 2004 (+ **10 jours**)
 - Publication par eEye des détails techniques sur la vulnérabilité : 18 Mars 2004
 - Exploitation par le vers Witty : 19 Mars 2004 (+ **1 jour**)

- Vulnérabilité dans le traitement des requêtes LDAP par Active Directory sous Windows 2000 (Windows 2000 SP4 puis MS04-011)
 - Rapportée par Core-ST à Microsoft : 16 Mai 2003
 - Corrigée par le SP4 de Windows 2000 : 26 Juin 2003
 - Publication des détails techniques par Core-ST : 2 Juillet 2003
 - **Re-découverte** d'une variante de la vulnérabilité par Core-ST : 11 Août 2003
 - Publication du correctif MS04-011 par Microsoft : 13 Avril 2004
 - <http://www.coresecurity.com/common/showdoc.php?idx=351&idxseccion=10>

- Vulnérabilité dans Internet Explorer (iLookup trojan)
 - Découverte "dans la nature" par un chercheur en vulnérabilité expert sur Internet Explorer : **6 Juin 2004**
 - Vulnérabilité utilisée par un éditeur de spyware bien connu
 - Confirme que les éditeurs de logiciels malveillants financent la recherche de vulnérabilités nouvelles affectant le poste client
 - <http://62.131.86.111/analysis.htm>
 - Exploitation via des serveurs IIS compromis via Scob/Download.Ject (code JavaScript) : **26 Juin 2004**
 - Serveurs IIS probablement compromis par une faille plus ancienne
 - Correctif MS04-025 publié par Microsoft : **30 Juillet 2004**

- Les vulnérabilités et codes d'exploitation (*exploits*) ont une valeur marchande certaine
 - Rémunération des découvreurs de vulnérabilités
 - Vulnerability Contributor Program (VCP) d'iDefense (<http://labs.iddefense.com/>)
 - Sociétés dont le modèle économique se construit sur les vulnérabilités et la mise à disposition d'exploits
 - Ex : ImmunitySec (<http://www.immunitysec.com/>), Core-ST (<http://www.corest.com>)
 - Malveillance et criminalité sur Internet sont intéressées par les vulnérabilités affectant les systèmes ou le poste client
 - Éditeurs de *spywares*, Spammeurs
 - Criminalité organisée (*phishing* notamment)
 - Gouvernements ou autres ayant un intérêt à utiliser des 0day pour mener des attaques

- Vulnérabilités systèmes
 - Ex : vulnérabilités dans les interfaces RPC des systèmes Windows
- Vulnérabilités du poste client
 - Multiples vulnérabilités des navigateurs web, lecteurs multi-média, ...
- Vulnérabilités dans des logiciels de sécurité
 - Vers Witty (mars 2004) exploitant une vulnérabilité dans le module de décodage d'ICQ des sondes de détection d'intrusion ISS
 - Vulnérabilités dans les antivirus F-secure, Symantec et Trend Micro découvertes par ISS (février 2005, Alex Wheeler et Neel Mehta)
- Vulnérabilités dans des applications web largement répandues
 - Vers Santy (décembre 2004) exploitant les forums phpBB, via des requêtes sur des moteurs de recherche (Google, AOL, Yahoo suivant les versions)
 - Récemment, Mailman, Awstats, phpBB, ...

- De plus en plus d'*exploits* publiés
 - Dans les listes électroniques usuelles
 - Via des boîtes à outils tel que Metasploit (<http://www.metasploit.org/>)
- Recherche de vulnérabilités et l'écriture d'*exploits* fait l'objet de plusieurs ouvrages et de nombreuses présentations publiques
 - *The Shellcoder's Handbook, Exploiting software : How to Break Code*
- Techniques nécessaires pour des *exploits* fonctionnels sont de plus en plus perfectionnées
 - Pour obtenir des *exploits* génériques pouvant fonctionner sur tous les systèmes
 - Pour contourner les restrictions mises en oeuvre au niveau des systèmes d'exploitation récents

Utilisations légitimes ou illégitimes d'exploits

- Les vulnérabilités et les *exploits* ont une valeur monétaire bien réelle
 - La recherche et découverte d'une vulnérabilité de type *0day* a une valeur évidente pour qui souhaite mener des actions malveillantes
 - Le développement d'un *exploit* fiable et générique nécessite une expertise forte, reposant sur des compétences pointues
 - Présentation *Analyzing exploit code quality* (Ivan Arce, Core-ST)
 - À la portée de certaines sociétés commerciales, de gouvernements, des malveillants ou du crime organisé
- Exemple typique de cette tendance : produit IMPACT de Core-ST
 - A l'origine, produit visant à industrialiser les tests d'intrusion
 - Boîte à outils d'*exploits* fiables, permettant de mener des tests d'intrusion
 - Présenté aussi comme outil complémentaire à d'autres solutions de sécurité
 - tester la bonne configuration de firewalls, l'efficacité d'IDS ou d'IPS
 - tester que les correctifs ont bien été appliqués, en validant que l'exploitation d'une vulnérabilité ne fonctionne plus

- Étapes classiques de la gestion des vulnérabilités
 - Répertorier les actifs logiciels du périmètre
 - Assurer une veille en vulnérabilités
 - Déterminer si ces logiciels sont vulnérables
 - Mettre en oeuvre des solutions pour mitiger les vulnérabilités
 - S'assurer que le risque a été correctement pris en compte et réduit

- Identification des actifs logiciels au sein d'un périmètre
 - Utiliser l'inventaire des équipements informatiques connus
 - Connait-on toutes les applications installées sur les serveurs ?
 - Des restrictions sur le poste de travail sont-elles en place pour empêcher l'installation de logiciels étrangers ?
 - Quid des autres équipements ?
 - Ex : imprimantes embarquant un serveur HTTP vulnérable, téléphones IP
 - Placer ces équipements dans des zones dédiées est-il suffisant pour ne pas se préoccuper des vulnérabilités ?
 - Les produits concernés sont-ils encore supportés par l'éditeur ?
- Préambule obligatoire à un processus de gestion des vulnérabilités
 - Lors de la publication d'une nouvelle vulnérabilité, élément clé pour déterminer rapidement l'impact et l'urgence de la prise en compte

- Rôles d'un service de veille en vulnérabilités
 - Surveiller les listes électroniques de sécurité typiques sur lesquelles sont annoncées les nouvelles vulnérabilités publiques
 - Une unique vulnérabilité donne typiquement lieu à de multiples messages
 - Ex : mise à jour pour une vulnérabilité annoncée par les différentes distributions Linux (RedHat, SuSE, Debian, Mandrake, Gentoo, Conectiva, Ubuntu, ...)
 - Plus généralement, une vulnérabilité largement répandue donnera lieu à de multiples bulletins de sécurité de la part des différents éditeurs
 - Ex : avis du CERT, lorsque la coordination de nombreux éditeurs est nécessaire avant la publication d'une vulnérabilité à large portée (vulnérabilités OpenSSL, LDAP, SNMP, ...)
 - Également les listes des éditeurs, lorsqu'il s'agit de vulnérabilités faisant l'objet de correctifs
 - Bulletins de sécurité Microsoft, HP, Sun, Apple, ...
 - Référentiel unique CVE (*Common Vulnerabilities and Exposure*) permet de faire le lien entre une vulnérabilité et les correctifs associés

- Lorsqu'un correctif est disponible
 - Évaluer la criticité de la vulnérabilité et statuer sur l'urgence d'appliquer ou non ce correctif
 - Qualifier le correctif et son impact sur l'existant
 - Déployer le correctif, à l'aide des solutions de type *patch management*
- Lorsqu'aucun correctif n'est (encore) disponible ou qu'il n'est pas possible/souhaitable de l'appliquer
 - Prendre des mesures permettant de limiter l'impact de la vulnérabilité
 - Nécessite des détails sur la façon dont la vulnérabilité peut être exploitée
 - Ces détails sont parfois donnés par l'éditeur lorsque le correctif est disponible
 - Si aucun correctif n'est disponible, il faut pouvoir comprendre la vulnérabilité et déterminer s'il est possible de la contrer
 - Ex : filtrage réseau renforcé, désactivation de la fonctionnalité vulnérable, ...
 - Ex : MS04-011 (création d'un fichier dcpromo.log en lecture-seule pour empêcher l'exploitation de la vulnérabilité LSASS)

- Audit de vulnérabilités
 - Avant : permet de déterminer les machines qui sont vulnérables, avant de se lancer dans la correction des vulnérabilités
 - Après : permet de s'assurer que les correctifs appliqués ou les mesures correctives sont bien effectives
- Approches possibles
 - Tester la présence du correctif associé à une vulnérabilité et considérer que le système est protégé si le correctif est appliqué
 - Approche adoptée par les scanners de vulnérabilités classiques tels que Microsoft MBSA ou Nessus
 - Effort OVAL, visant à référencer les éléments permettant de déterminer la présence d'une vulnérabilité
 - Tenter d'exploiter la vulnérabilité testée, afin de s'assurer que le correctif fait bien son travail
 - Approche mise en avant par Core-ST avec son produit IMPACT

- IPS (Intrusion Prevention Systems)
 - *Network-based IPS* : un NIPS bloquera t-il des tentatives d'exploitation d'une vulnérabilité ou un vers en train de se propager ?
 - *Host-based IPS* : ensemble de logiciels (agents) installés sur un système assurant un durcissement global et la détection/blocage de fonctionnalités typiquement exploitées par des logiciels malveillants
- Firewall (personnels)
- Antivirus
 - L'antivirus saura détecter les fichiers déposés par un vers/virus mais n'empêche pas l'exploitation d'une vulnérabilité
- Ne pas oublier de durcir les systèmes avant d'y ajouter des couches de sécurité supplémentaires
- Une architecture réseau bien conçue doit permettre de limiter l'impact de l'exploitation d'une vulnérabilité

- La recherche et l'exploitation de vulnérabilités logicielles sont des secteurs porteurs
 - De nombreux et divers intérêts motivent la recherche de vulnérabilités logicielles
- Face aux nombreuses vulnérabilités publiées, un travail de qualification est impératif
 - Avoir une idée des scénarios possibles d'exploitation permet de mieux cerner les risques associés
 - Une vulnérabilité à large portée (ex: affectant les systèmes Windows par défaut) a de nombreuses chances de se retrouver dans un vers se propageant à large échelle
- Solutions autour de la gestion des vulnérabilités
 - IPS (Intrusion Prevention Systems), suite de sécurité du poste de travail (firewall personnel, anti-virus), contrôle des accès au réseau avec 802.1X, ...

- Listes reprenant les listes de sécurité classiques
 - Secunia : <http://www.secunia.com/>
 - Seifreid-security : <http://www.seifreid.org/security/>
 - Securiteam : <http://www.securiteam.com/>
- Bases de vulnérabilités
 - CVE (Common Vulnerabilities and Exposure) : <http://cve.mitre.org/>
 - ICAT Metabase : <http://icat.nist.gov/>
 - OVAL (Open Vulnerability and Assessment Language) : <http://oval.mitre.org/>
 - OSVDB : Open-Source Vulnerability Database : <http://www.osvdb.org/>
 - Symantec vulnerability database : <http://www.securityfocus.com/bid/>
 - ISS X-Force database : <http://xforce.iss.net/xforce/search.php>

- Tendances des vulnérabilités
 - SANS Internet Storm Center (ISC) : <http://isc.sans.org/>
 - Blog de F-secure (actualité sur les nouveaux virus)
 - <http://www.f-secure.com/weblog/>
 - SANS Top 20 vulnerabilities : <http://www.sans.org/top20/>
 - Symantec Internet Security Threat Report
 - <http://enterprisesecurity.symantec.com/article.cfm?articleid=4776> (résumé)
 - <http://enterprisesecurity.symantec.com/content.cfm?articleid=1539>
- Gestion des correctifs de sécurité
 - <http://www.patchmanagement.org/>