netmonk security blog

In real life, i'm a computer (in)security specialist.

mercredi, mars 15, 2006

Virus and RFID

Imagine an Rfid chip in every chicken, Imagine this rfid contains a virus, Imagine that each time this chip is "read" by human computer, it infects it. Well this is not a new sci-fi fantasm, this is real. First chicken gots Birdflue, now they can get rfid virus.

More information on this serious and official website about rfid virus

posted by netmonk : 3/15/2006 03:40:00 PM

jeudi, janvier 19, 2006

A little piece of my bookmark without any order

```
### Just a little sed command for extracting and cleaning
$home/.mozilla/firefox/bookmark.html
### just copy bookmark.html to $home under bookmark.old name before
### sed "s/ADD.*\">/>/g" bookmark.old >bookmark.html
```

Casting
FrSIRT 24/24 & 7/7 - Centre de Recherche et de Veille en SÃ©curitÃ© Informatique
InfoSecDaily / News
HijackThis Log file
Serveur SuperMicro et Transit IP
Knoppix STD
University of Mannheim- Lehrstuhl Praktische Informatik I index
Offensive Computing | Malicious code, its whats for dinner!
Noam Chomsky
recent news headlines
17 Mistakes Microsoft Made in the Xbox Security System - Xbox-Linux
InfoHacking
Damn Small Linux (DSL), 50 megabytes of penguin power
Support Information
Ars Technica - The PC Enthusiast's Resource
VOID.AT Security
PullThePlug.org
Index of /
gavare.se: GXemul: History and Future
EUCD.INFO : sauvons le droit d'auteur !
LibertÃ©s numÃ©riques
55k7 // news
Watchfire Security Zone
ipLegion
What is CMMI?
ADD / XOR / ROL
http://www.ietf.org/rfc/rfc2326.txt
http://www.cgisecurity.com/papers/fingerprinting-2.html
IT Security by girls.
Foofus Networking Services - Medusa
Computer Viruses - Theory and Experiments
Helbreath Top 50 v. 6.0.4 - Its already been a year! - Main Page - All Sites
chargen 19/udp
del.icio.us
hebergement site internet et serveur dedie - OXYD
Windows Security: Vulnerabilities in Graphics Rendering Engine May Still Exist

Contact

plonky at gmail dot com

Even After Applying the MS05-053 (KB896424) Security Update
hexale
gera's InsecureProgramming page
US-CERT Cyber Security Bulletin SB2005 -- Cyber Security Bulletin 2005 Summary
glish.com : CSS layout techniques
Networksecurity.fi - Security Research - Juha-Matti Laurio, Finland
netmonk --=[Enter The Church]=--
Forum
Uninformed
Vulnerabilite.com - SÃ©curitÃ© informatique

Sed - An Introduction and Tutorial

Index of data/
Luttes Etudiantes
JOOReports
FreeIX - Members
Default Password List
Welcome To ^C^ rime Mechine
Apollo Masters Shopping Cart
The Secret Society of Lathe Trolls :: Index
Information
Welcome to Vinylium
SH Forums - powered by vBulletin
Record Cutting :: Index
milw0rm.com
Digital Armaments for Intelligence
http://www.phlak.org/tools.htm

Enjoye

posted by netmonk : 1/19/2006 12:50:00 PM

mercredi, janvier 18, 2006

back to the roots

Links

Google News
My buddy train's blogs
securite.org
Marek Bialoglowy's blog
Wushu/kungfu ressources

Archives

juillet 2005
janvier 2006
mars 2006

hey, after some uninteresting work, a question raised in my mind: what is the first buffer overflow released publicly in the world ?

i think i have the answer here :
http://seclists.org/lists/bugtraq/1995/Dec/0002.html
Note the date its 1995 :)
11 years old, damn.

posted by netmonk : 1/18/2006 11:37:00 AM

mercredi, janvier 04, 2006

les joies du shell scripting

Introduction

Le shell est un outil fondamental dans les systèmes Unix. Il sert principalement d'interface entre l'utilisateur et le système d'exploitation. C'est lui en particulier qui s'occupe d'exécuter les commandes que l'utilisateur saisit. Mais sa fonction ne s'arrête pas là, il
offre à l'utilisateur toute une palette de service, tel que la possibilité
de manipuler les processus comme par exemple les communications par le
biais des tubes, de manipuler des fichiers en utilisant les redirections.
Il offre aussi un environnement d'exécution qui affecte aussi bien son
comportement que le comportement des processus dont il a la charge.

De part la diversité des systèmes Unix, et l'ancienneté de ces systèmes, il est normal que cela se ressente dans la variété des shells qui
sont disponibles. Au fil des ans, certains shells se sont distingués, le

korn shell (ksh), les shells tcsh et csh, le bourne shell (sh) et son poussin le bash. Il existe d'autres shells mais leur diffusion est limitée.
Une caractéristique des shells est de posséder un langage de script. Un
langage de script se distingue des langages de programmation habituels du
fait qu'un script n'est pas transformé en exécutable binaire autonome qui
contient du code machine, mais a besoin d'un programme qui interprète le
script à chaque exécution de celui-ci.

Comme le shell a pour vocation d'aider l'utilisateur dans la gestion des fichiers et des processus, les langages de script associés sont
particulièrement adaptés à cet usage. L'écriture de script pour faciliter
l'administration système est très courant dans l'industrie, qui bien souvent tend à minimiser les coûts humains de maintenance et gestion de ces
systèmes.

Dans la suite de ce document, nous nous attacherons particulièrement au langage de script 'bourne shell' (sh) qui est installé
par défaut sur tout système Unix. Ce shell d'ailleurs ne présente pas beaucoup d'intérêt pour une utilisation interactive, mais le fait qu'il soit installé en standard sur tout type de système Unix, garantie la portabilité des scripts.

Pour écrire un shell script il n'est pas besoin d'outil
spécifique, seul votre éditeur de texte préféré fera l'affaire (emacs, vi ,
.).

Il existe deux manières d'invoquer un script shell. La première est d'exécuter le shell dans lequel il est écrit, avec comme argument le

nom du
script (par exemple : sh monscript.sh). La deuxième manière consiste d'une
part à rajouter au fichier comme première ligne un appel au shell dans
lequel le script est programmé en commençant la ligne par ' # !' et en
fournissant ensuite le chemin vers l'exécutable du shell en question (dans
notre cas cela donne : # !/bin/sh). Ensuite il suffit de rendre exécutable
le script au yeux du système à l'aide de la commande chmod. Il suffit
d'invoquer le script par son nom dans la ligne commande pour l'exécuter
(par exemple ./monscript.sh).

Le langage sh

Le langage du shell sh possède des structures de control tel que
IF..THEN..ELSE, FOR, WHILE. Il fournit plusieurs moyens d'accéder aux
variables en lecture et en écriture. Il propose aussi une extension des
possibilités de redirection, et un certain nombre de commande interne.

Création de variable

La syntaxe pour assigner une variable est la suivante : nom=valeur. Si
cette variable n'existait pas auparavant, elle est de facto crée, sinon
l'ancienne valeur est écrasée. Une variable fraîchement crée est toujours
une variable locale. C'est à dire qu'elle appartient au contexte du shell
en cours d'exécution et à ses processus fils. Pour assigner une valeur qui

contient des espaces, il suffit d'englober la valeur autour de double guillemet (par exemple toto="toto ta"). Il est important qu'il n'y ai pas
d'espace entre '=' et la valeur.

Pour accéder aux variables, il existe plusieurs moyens :

. $nom est remplacé par la valeur de la variable

. ${nom} est remplacé par la valeur de la variable, cela est utile lorsque la chaîne de caractère est immédiatement suivie par une autre
chaîne (${poids}kg, affichera 17kg si poid=17)

. ${nom-mot} est remplacé par la valeur de nom si la variable est initialisée et par mot sinon

. ${nom=mot} assigne la valeur mot à nom si cette variable n'est pas initialisée et est remplacé ensuite par la valeur de nom.

. ${nom ?mot} est remplacé par la valeur de nom si cette variable est
initialisée et sinon renvoie mot sur la sortie d'erreur et le script se termine. Si mot est vide alors un message d'erreur standard est affiché.

Pour initialiser une variable à partir de l'entrée standard (généralement
le clavier ou une redirection) il faut utiliser la commande read. Elle assigne successivement les mots entrés aux noms de variables donnés en
argument. Si le nombre de mot est supérieur aux nombre de variable, alors
la dernière variable écope du surplus de mot.

Pour exporter une variable locale à l'environnement, il faut utiliser la
commande export (par exemple export nom=toto). La commande

env permet
d'afficher le contenu de l'environnement.

Pour rendre une variable nom modifiable, il suffit d'utiliser la commande
readonly, avec le nom de la variable en argument.

Le shell sh fournit un certain de variable locales prédefinies.

. $@ est la liste des paramètres de la dernière commande exécutée

. $# est le nombre de paramètre de la dernière commande exécutée

. $? est la valeur de sortie de la dernière commande exécutée

. $$ est le PID du processus en cours d'exécution (généralement le shell qui interprète le script.

Le shell sh possède un outil d'évaluation d'expression arithmétique nommé
expr. Il supporte une variété d'opérateur : */%, +-, => >= < <= !=, &, | .
Il supporte aussi des opérations sur les chaînes de caractères comme :
match, substr, index, lenght.

Le shell sh possède un outil de test d'expression judicieusement appelé
test. Une liste non exhaustive des options est donnée :

. -f nomdefichier, teste si le fichier nomdefichier existe

. -l chaîne, teste sur la chaîne est non vide

. -n chaîne, teste si la chaîne a au moins un caractère.

. -z chaîne, teste si la chaîne est vide

. str1 = str2 est vrai si str1 est égal à str2

. str1 != str2 est vrai si str1 est différent de str2

. int1 -eq int2 est vrai si int1 est égal à int2

. int1 -ne int2 est vrai si int1 est différent de int2

. int1 -gt int2 est vrai si int1 est plus grand que int2

. int1 -ge int2 est vrai si int1 est plus grand ou égal à int2

. int1 -lt int2 est vrai si int1 est plus petit que int2

. int1 -le int2 est vrai si int1 est plus petit ou égal à int2

. !expr est vrai si expr est faux

. expr1 -a expr2 est vrai si exrp1 et expr2 sont vrais

. expr1 -o expr2 est vrai si expr1 ou expr2 sont vrais.

Les structures de contrôle

La première structure est case .. in .. esac, qui permet plusieurs branchements suivant la valeur d'une variable.

Par exemple :

```
case $toto in

"1")

date

;;

"2")
```

```
pwd

;;

*)

echo choix illégale

;;
```

La structure for.. do ..done permet de faire des traitement sur une variable qui va prendre des valeurs successives, ces valeurs étant fournit
par une liste si cette variable n'est pas déjà une liste.

Par exemple

```
for i in *.zip

do

gunzip $i

done
```

La structure if ..then .. fi permet de faire des branchements conditionnels
suite à un ou plusieurs tests.

Par exemple :

```
if test $nombre -eq 0

then echo 0
```

else

echo non zero

fi

Il est possible de faire un test de manière différente, nous aurions pu
écrire '[$nombre -eq 0]' à la place de 'test $nombre -eq 0'.

La structure while ..do ..done évalue une série de commande et exécute une
deuxième série de commande si la première a réussi.

Par exemple

while [$x -le $1]

do

echo $x

x=`expr $x + 1`

done

Voici en teneur les spécificités du langage de script sh. Englober une expression par le caractère ` provoque l'évaluation de cette expression.

posted by netmonk : 1/04/2006 04:44:00 PM

Glanage d'information

Lors d'un audit recent, la necessite d'identifier une application web s'est faite

ressentir. L'idee n'est surement pas nouvelle, mais les outils sont inexistants. En fait l'idee principale qui me motivait etait de trouver la societe qui a techniquement vendu a mon client, le site en question.

Aujourd'hui le monde du developpement web c'est tres fortement rationnalise, fini le temps du developpement from scratch. Les societes de service utilisent des applications web generiques, qu'elles adaptent tres rapidement au besoin de leur client, l'aspect design etant parfois traite par un autre prestataire. Tout ceci principalement pour des raisons de cout, induit par la reutilisation d'outil majoritairement achetes a des societes specialises.

Des lors, il peut etre interressant d'identifier les prestataires en analysant les technologies vendues. Une cartographie des relations entre entreprise et prestataire peut etre ainsi etablie. Cela peut etre interressant d'un point de vue securite (une faille chez l'un signifie une faille chez l'autre), d'un point de vue marketing et guerre de l'information (recolte d'information et analyse des relations).

Voici un exemple, base sur l'experience de l'audit. La structure des URLs etait tres frappante a mes yeux, j'ai alors cherche a connaitre quels etaient les autres sites qui utilisaient la meme structure d'url(resultats ici)

Il y a deux choses surprenantes :
- il y a peu de reponses
- la majorite des reponses provient de site francais lies a l'administration ou a des ministeres francais.

Je laisse au lecteur le soin de trouver le presume prestataire.

posted by netmonk : 1/04/2006 10:52:00 AM

mardi, janvier 03, 2006

Microsoft et la faille WMF

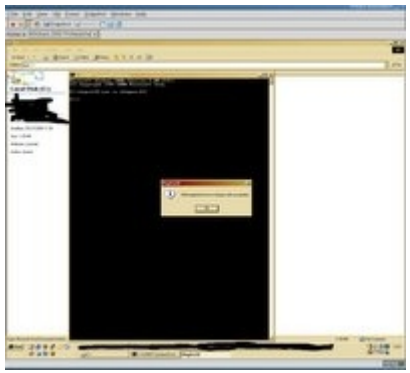
La semaine derniere, plusieurs sites web diffusant un contenu un peu special ont ete rapportes sur diverses listes de diffusion (bugtraq, dailydave ...).

Lorsque vous ouvriez l'adresse de ces sites, un fichier portant l'extension WMF etait propose automatiquement en telechargement.

WMF signifie Windows Meta File, que Windows utilise pour stocker des images au format bitmap ou vectoriel. Une des particularites du format vectoriel est qu'il contient des commandes qui servent au rendu graphique. Ces commandes sont interpretees et le resultat affiche par la fonction PlayMetaFile() de l'API Windows.

Le probleme de securite vient de l'etape d'interpretation des commandes lors de la generation de l'apercu dans l'explorateur windows. La faille permet alors d'executer n'importe quelle commande avec les droits de l'utilisateur qui est victime.

Le principal soucis c'est qu'un fichier MetaFile n'a pas besoin de porter l'extension .wmf pour etre considere comme tel. Il peut tres bien porter l'extension .jpg, .tif, .png, .gif...
De plus il ne suffit pas de desactiver le mode apercu dans l'explorateur windows pour ne plus etre vulnerable. Au contraire. Windows continue de generer des apercus des fichiers en interne au cas ou.



Le remede le plus simple est de desactiver temporairement la librairie shiwgvw.dll en tapant la commande suivante dans cmd.exe:
c:\regsvr32.exe /u shimgvw.dll
Et de clicker sur OK dans la boite de dialogue qui s'affiche alors.

Il faut attendre patiemment un patch Microsoft, sachant qu'a l'heure ou j'ecris aucun patch microsoft ne comble la faille. A noter que tous les OS Microsoft semble affectes par cette faille, et il semblerait qu'il en soit de meme pour Lotus Domino d'IBM

posted by netmonk : 1/03/2006 01:16:00 PM

vendredi, juillet 29, 2005

Welcome here

Hello dear readers,

After a little talk with a mate while we were going to work using the suburban train. I decided to reactivate this little place which gives me the opportunity to share my idea, my critics, my thoughs, and so on.

Regards

posted by netmonk : 7/29/2005 02:00:00 PM 1 comments