FICHIER		VULNERABILITE	PATCH	
NOM	**#ID**	**DESCRIPTION**	**DESCRIPTION**	**patché ?**
daemon.c	#1	Ligne 123 : le fichier pid n'est pas protégé en écriture. Il est donc possible de modifier le pid lut par la fonction et ainsi utiliser les droits utilisateurs du serveur pour mettre hors service l'infrastructure par kill.	Utiliser un chmod 600 lors de la création du fichier pid	x
main.c	#2	Ligne 63 : buffer overflow potentiel sur un password d'une taille > 64. Le buffer userPassword est de 512 octets et celui de savePassword de 64	Utiliser un strlcpy/strncpy ou placer un \ 0 dans la chaine password.	x
	#3	Ligne 61 : pas de vérification de password : pourrait etre NULL (NULL dereference sans vérification)	Verifier password!= NULL avant de continuer	x
	#4	Ligne 66 : Integer overflow sur un mot de passe trop long.	utiliser un unsigned int	x
	#5	Ligne 88 : Buffer overflow sur userPassword		x
	#6	Ligne 101 : Buffer overflow sur adminPassword		x
	#7	Ligne 278 : Pas de borne supérieur à money (integer overflow)	Vérifier que money n'est n'atteint pas la taille max d'un int – 10	x
	#8	Ligne 315 : Buffer overflow sur log (char [128]) par sprintf	Vérifier la taille de ingredientName	x
	#9	Ligne 317 : Appel système peut être détourné via la modification des variables d'environnement.	Utiliser write pour écrire dans le fichier log	x
	#10	Ligne 350 : getNumber peut retourner un négatif. Donc accès à la mémoire précedent le tableau handlerTab.	Vérifier le retour de getNumber	x
	#12	Ligne 71 : heap overflow (malloc peut retourner null, non vérifié donc écriture au début du prog)	Vérifier le retour de malloc	x
	#13	Mot de passe en clair dans le code	Utiliser un fichier de configuration protégé en lecture/écriture à lire au lancement u programme.	non patchable car modifie le fonctionnement global du serveur
network.c	#11	Ligne 74 : Format string : fprintf prend en parametre msg qui peut être formaté avec des flags et donc accéder à une mémoire interdite	Utiliser fprintf(err, "%s", msg);	x

poulet_a:ghukas_g:malbra_t:broggi_t

rendu