Ca sent le SAPin !
Voyage dans le monde d'un chercheur en sécurité SAP

Yvan Genuer
yvan.genuer@devoteam.com

Devoteam

Résumé. Spécifique, propriétaire, compliqué, gros, cher, usine à gaz... Voici certains des adjectifs que nous entendons souvent dès que nous abordons le sujet de la sécurité SAP. Cet article a pour but de montrer, par un exemple concret, que les préjugés sur SAP ne sont pas tous fondés. Analyse réseau, création d'outils open source et ingénierie inversée sur un langage propriétaire, tels sont les sujets abordés qui aboutissent à la découverte de vulnérabilités triviales.

1 Introduction

1.1 Qu'est-ce que SAP ?

Avec plus de 365 000 clients dans 180 pays, SAP est le leader mondial des ERP. 87 % des plus grosses entreprises mondiales utilisent SAP [4]. Les solutions SAP Netweaver sont des systèmes propriétaires et normalement fermés. La sécurité sur SAP a changé ces dernières années et une poignée de chercheurs ont mis au jour des centaines de vulnérabilités sur différents composants. Cela va du simple XSS très classique à l'exploitation spécifique d'une mauvaise configuration. Malgré tout cela, il reste encore des zones d'ombre sur les systèmes SAP.

1.2 Architecture d'un système SAP

La figure 1 représente de façon — très — simplifiée un système SAP Netweaver. Classiquement les utilisateurs utilisent le SAPGui, un client SAP à installer sur les postes de travail pour s'authentifier sur le système et travailler dessus. C'est le ABAP Dispatcher qui est chargé de répartir la charge des requêtes utilisateurs dans les Work Processes. Ces processus, visibles sur le système d'exploitation, effectuent le travail demandé et peuvent communiquer avec la base de donnée.

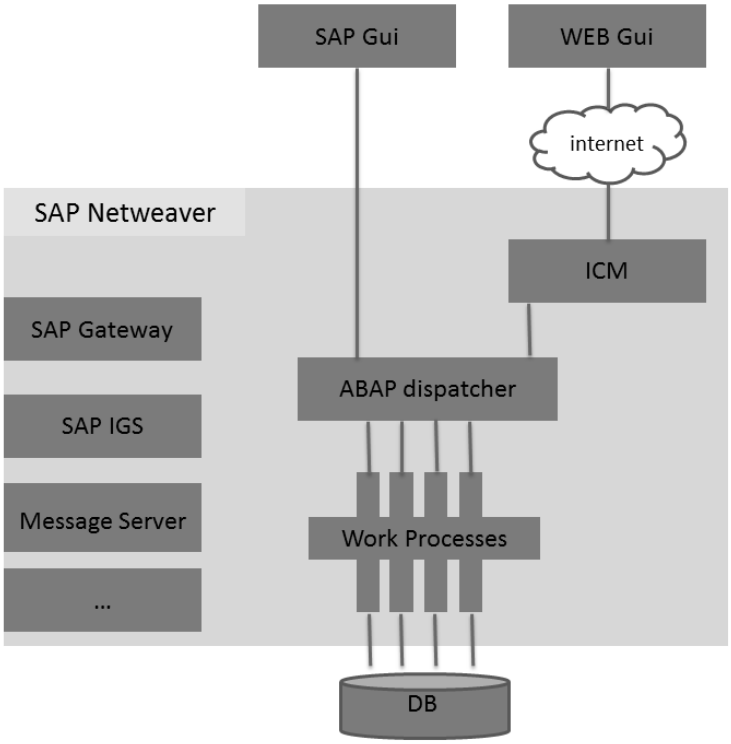

Fig. 1. Architecture de SAP

Un système SAP a besoin de plusieurs services pour fonctionner. Il en existe beaucoup, d'où le surnom d'usine à gaz de ces systèmes. Les plus connus sont le service SAP Gateway qui est chargé des communications externes, ou l'ICM qui est en fait un serveur web, ou encore le SAP IGS, service que nous allons étudier dans cet article.

Enfin SAP utilise un vocabulaire pouvant prêter à confusion. Voici donc un mini lexique aidant à comprendre la suite.

SID Sap IDentifier. Choisi à l'installation, cet identifiant sur trois caractères sert à désigner un système SAP.

`<sid>adm` Le nom de l'utilisateur administrateur sur le système d'exploitation. Il est dérivé du SID, par exemple `bobadm`, `prdadm`, `d01adm`.

Numéro de système Un identifiant sur deux chiffres, renseigné durant l'installation. Il sert à identifier un système SAP, mais surtout il définit les ports d'écoute par défaut du système. Par exemple, pour un SN=05, le port de la SAP Gateway sera 3305, celui de l'ICM 8005, etc.

ABAP Advanced Business Application Programing. C'est le langage propriétaire de SAP.

Report Programme ABAP.

Module fonction Fonction ABAP, qui peut être appelée indépendamment.

Transaction Sorte d'alias permettant de lancer une séquence de report (`SU01`, `DB02`, `SM21`, `SMGW`, etc).

1.3 Concepts importants pour cet article

Il est important d'expliquer quelques concepts propres aux systèmes SAP Netweaver, pouvant parfois heurter la sensibilité des plus jeunes.

- Sur les systèmes d'exploitation linux/Unix les services SAP tournent sous l'utilisateur administrateur <sid>adm...
- Ce que l'on appelle le noyau SAP est en fait un répertoire contenant les binaires, environ 500 Mo, utilisés par le système SAP. Il existe un noyau SAP par type d'architecture, de version et de base donnée. Le code source de ce noyau n'est pas disponible.
- En revanche, le code source de l'ensemble des reports ABAP est stocké dans la base de données. Pourquoi ? Car ces sources ABAP sont compilées par le noyau SAP, puis stockées dans la base de donnée. Ce niveau d'isolation par rapport au système d'exploitation permet aux programmes ABAP d'être portables sans modifications sur tous les systèmes supportés par SAP. En contrepartie il est possible d'accéder à ces sources via le SAPGui.
- Enfin, un système SAP Netweaver intègre, par défaut, un environnement complet de développement ABAP. Cela permet de créer, d'éditer et même de debugger du code ABAP et ceci quelle que soit la fonction du système SAP : production, développement, formation, test...

2 SAP IGS

2.1 Fonction du service SAP IGS

SAP Internet Graphic Server (IGS) est un composant présent par défaut dans les systèmes SAP Netweaver depuis 2005, avec la version 6.40. Il fournit au système différents services tels que la génération de graphiques ou de cartes, la conversion d'images ou encore la compression de fichiers. Typiquement, quand le système SAP crée un rapport avec de jolis graphiques, c'est l'IGS qui a généré ces graphiques.

Il est possible d'accéder à ce service depuis l'extérieur via HTTP(S) ou via un protocole propriétaire SAP, ceci pour permettre à d'autres systèmes SAP ou à des outils tiers d'utiliser ce service. En interne, dans

le système SAP lui-même, la communication se fait via des sockets locales et seulement via le protocole propriétaire.

L'administration de l'IGS se fait avec la transaction SIGS, depuis un SAPGui, avec un utilisateur SAP à privilèges.

2.2 État de l'art

Il existe une documentation sur SAP IGS, mais elle ne concerne que les parties installation, configuration et administration [5]. Il n'existe pas de documentation sur les spécifications du protocole utilisé ou sur le fonctionnement des différents processus qui le composent.

En mars 2018, les premiers résultats de mes recherches sur le sujet ont été présentés [10]. Plusieurs informations importantes concernant ce service y sont expliquées. Cet article s'inscrit dans une continuité en abordant le sujet du protocole propriétaire utilisé par ce service.

2.3 Architecture d'un IGS

Le service SAP IGS s'articule sur plusieurs composants, comme montré dans la figure 2 :

- **Multiplexer**. Processus qui gère les communications et répartit les demandes vers les Portwatcher. Il écoute sur les ports 4xx00 et 4xx80, respectivement pour RFC et HTTP, où xx est le numéro du système SAP, sur deux chiffres.
- **Portwatcher**. Processus qui traite la demande. Chaque processus écoute sur un port unique, 4xxyy, où yy est l'identifiant du processus sur deux chiffres et xx toujours le numéro du système SAP.
- **Interpreter**. Type de demande chargée dans dans le Portwatcher. Par exemple `IMGCONV` pour la conversion d'image. Les demandes de type administration (`ADM :*`) sont chargées par défaut.

3 Le protocole réseau IGS

3.1 Schéma des tests

Dans le cas d'un système SAP, sur Linux, les Work Process utilisent des sockets Unix pour communiquer avec le service IGS. Ces sockets sont décrites dans le tableau 1.

L'interface d'administration de l'IGS permet de tester les différents interpreters, d'obtenir un status sur les portwatchers ou encore de lire des

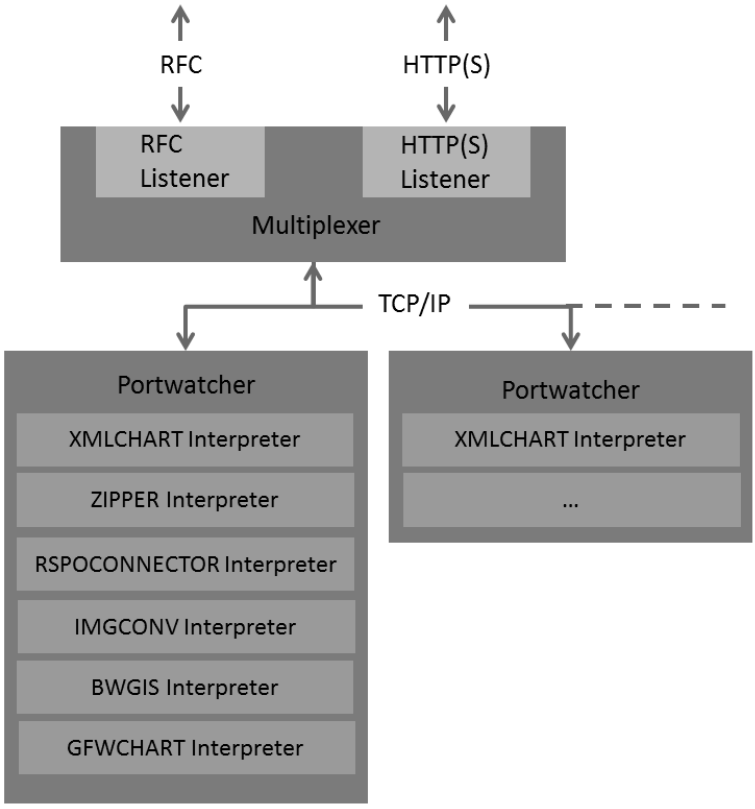

Fig. 2. Architecture du service SAP IGS

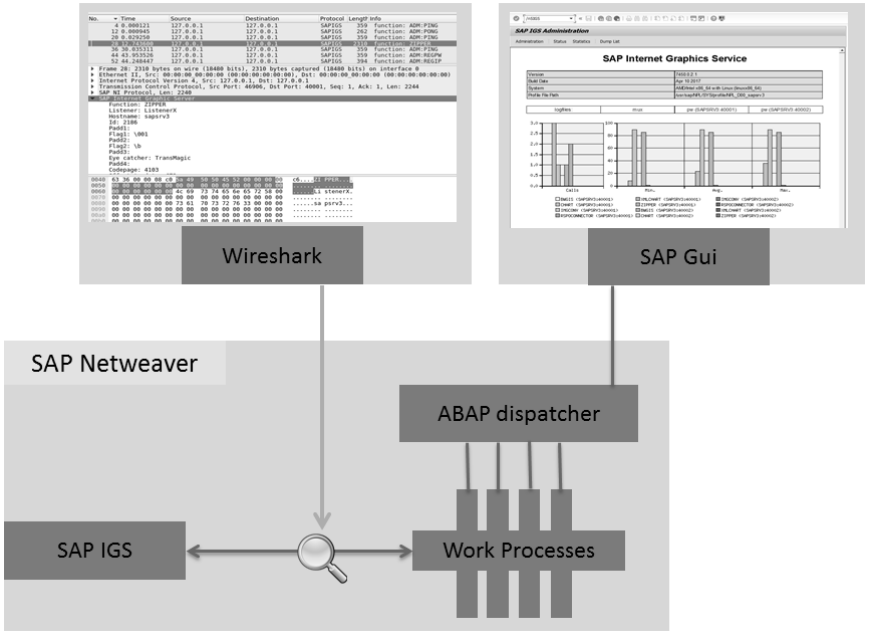

Fig. 3. Schéma des tests

fichiers de logs. Pour analyser les échanges avec le service IGS, il suffit donc de générer du trafic à partir de l'interface d'administration et de capturer les communications sur ces sockets, comme le montre la figure 3.

3.2 Résultat

1. Les premières constatations montrent que le protocole n'est pas offusqué ou sécurisé. Les champs sont en clair et facilement interprétables, voir la figure 4.

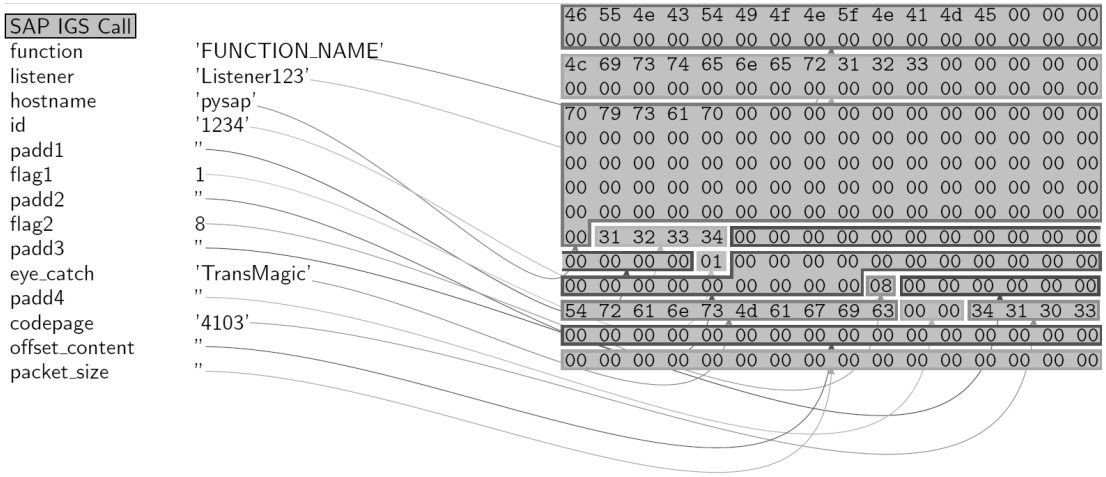

Fig. 4. En-tête paquet SAP IGS

2. Dans le noyau SAP, les binaires associés à ce service ne sont pas strippés. Une étude rapide permet de ressortir l'ensemble des fonctions possibles, listées dans le tableau 2.
3. Il n'y a pas de mécanisme d'authentification. Comme la communication se fait normalement depuis un Work Process en interne, le service lui fait confiance. En effet, pour utiliser ces Work Process, il faut passer par le SAPGui et donc être authentifié.
4. Finalement, les tests ont montré qu'il est possible de communiquer directement avec le service IGS, sans passer par le SAPGui. Le Multiplexer fait suivre la requête à un Portwatcher qui l'exécute et retourne le résultat au Multiplexer. En revanche ce dernier ne sait pas d'où vient la demande initiale et termine la communication sur une erreur. Une communication directe depuis l'extérieur avec l'IGS se fait donc donc en aveugle, mais permet tout de même à notre requête d'être exécutée, sans être authentifiée.

/tmp/.sapstream40000	socket du multiplexer
/tmp/.sapstream40001	socket du portwatcher 1
/tmp/.sapstream40002	socket du portwatcher 2

Tableau 1. Socket local de l'IGS

Nom	Information
ZIPPER	Compression de fichie
IMGCONV	Conversion d'image
RSPOCONNECTOR	Information sur les imprimantes
XMLCHART	Génération de graphique
CHART	Génération de graphique
BWGIS	Génération de carte
SAPGISXML	Génération de carte
ADM:REGPW	Enregistrement d'un Portwatcher
ADM:UNREGPW	Désenregistrement d'un Portwatcher
ADM:REGIP	Enregistrement d'un Interpreter
ADM:UNREGIP	Désenregistrement d'un Interpreter
ADM:FREEIP	Informe que l'Interpreter est disponible
ADM:ILLBEBACK	Renvoie le résultat de la requête
ADM:ABORT	Arrêt d'une requête
ADM:PING	Ping reçu
ADM:PONG	Ping envoyé
ADM:SHUTDOWNIGS	Arrêt du service IGS
ADM:SHUTDOWNPW	Arrêt d'un Portwatcher
ADM:CHECKCONSUMER	Vérifie l'état d'un Portwatcher
ADM:FREECONSUMER	Informe qu'un Portwatcher est disponible
ADM:GETLOGFILE	Affiche des fichiers de logs
ADM:GETCONFIGFILE	Affiche la configuration
ADM:GETDUMP	Liste des fichiers de dump
ADM:DELETEDUMP	Supprime un fichier dump
ADM:INSTALL	Voir la partie 5
ADM:SWITCH	Augmente ou diminue le niveau des traces
ADM:GETVERSION	Retourne la version de l'IGS
ADM:STATUS	Affiche le status de l'IGS
ADM:STATISTIC	Affiche des statistiques
ADM:STATISTICNEW	Affiche des statistiques
ADM:GETSTATCHART	Affiche des statistiques en graphiques
ADM:SIM	Simulation

Tableau 2. Fonctions IGS

4 Contribution et outillage

A partir de ces résultats, j'ai pu développer des outils, mais surtout ajouter ce nouveau protocole à des outils existants.

4.1 PySAP

PySAP [1] est une librairie Python fournissant des modules pour créer et envoyer des paquets en utilisant les protocoles de SAP. Plusieurs protocoles sont déjà intégrés, comme NI, Router, MS, ENQ. J'ai donc ajouté le protocole IGS (voir listing 1).

```
>>> from pysap import SAPIGS
>>> dir(SAPIGS)
['ByteEnumKeysField', 'ByteField', 'ConditionalField', 'EnumField',
    'IP6Field', 'IPField', 'IntField', 'Packet', 'Request', 'SAPIGS'
    , 'SAPIGSTable', 'SAPNI', 'SignedShortField', 'StrFixedLenField'
    , 'StrFixedLenPaddedField', 'TCP', '__builtins__', '__doc__', '
    __file__', '__name__', '__package__', 'bind_layers', '
    igs_req_adm', 'igs_req_interpreter']
```

Listing 1. Module SAPIGS pour PySAP

4.2 SAP-Dissection

SAP-Dissection [2] est un plugin Wireshark supportant les protocoles SAP comme NI, Router, DIAG, ENQ. Là aussi, j'y ai ajouté le protocole IGS (voir figure 5).

5 Focus sur la fonction `ADM:INSTALL`

5.1 Recherche d'information

Cette fonction est listée dans les binaires du service, mais n'est jamais appelée lors de l'utilisation de l'interface d'administration de l'IGS. C'est simplement à cause de son nom que j'ai décidé d'investiguer de plus près sur elle...

N'étant pas documentée non plus, j'ai commencé par rechercher si elle était présente dans le code source ABAP d'un SAP Netweaver. En utilisant le programme `RS_ABAP_SOURCE_SCAN`, sur l'ensemble des class, includes et module, la chaine « `ADM:INSTALL` » apparait dans une méthode nommée `SAPMAP_INSTALL_SHAPEFILES`, visible dans la figure 6.

Cette méthode a pour but de copier des fichiers shapefiles manquant dans l'IGS. Un shapefile ou fichier de forme est un format de fichier pour

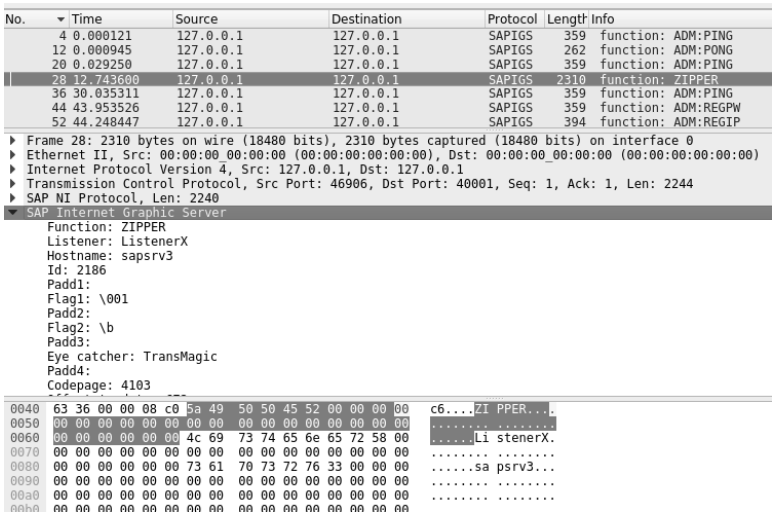

Fig. 5. Plugin Wireshark

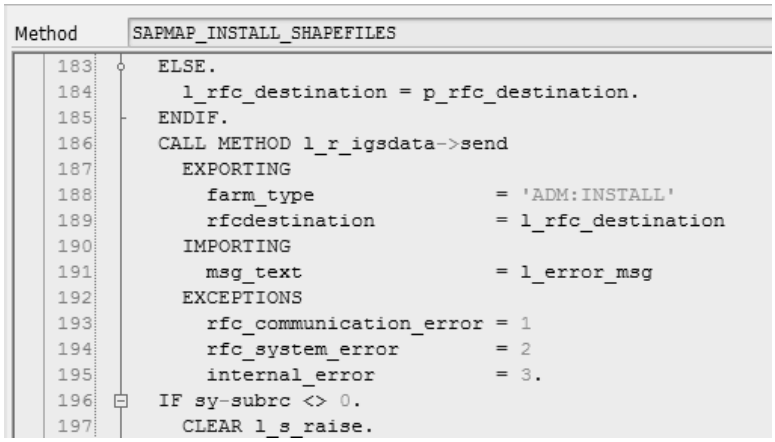

Fig. 6. `ADM:INSTALL` dans la méthode `SAPMAP_INSTALL_SHAPEFILES`

les systèmes d'informations géographiques (GIS). Dans SAP IGS, ces fichiers sont utilisés par l'interpreter BWGIS, interpreter qui génère des cartes géographiques afin par exemple de présenter les dépendances entre les données commerciales et spatiales.

Ce programme ABAP vérifie donc si il y a des shapefiles manquants. Si oui il prépare une table dans laquelle il entre les informations (nom, taille), ainsi que le contenu des fichiers à envoyer à l'IGS. Trois fichiers sont attendus, un .shp, un .dbf et un .shx. Enfin il envoie cette table à l'IGS via la fonction `ADM:INSTALL`.

Il serait donc possible de téléverser des fichiers directement sur le système SAP en utilisant la fonction `ADM:INSTALL` du service IGS. Pour le vérifier nous allons analyser cette méthode avec le debugger ABAP.

5.2 ABAP Debugger

Cette méthode ABAP étant statique il est possible de la tester directement. J'utilise ici le debugger intégré à tous les systèmes SAP via la transaction `se80`, voir figure 7.

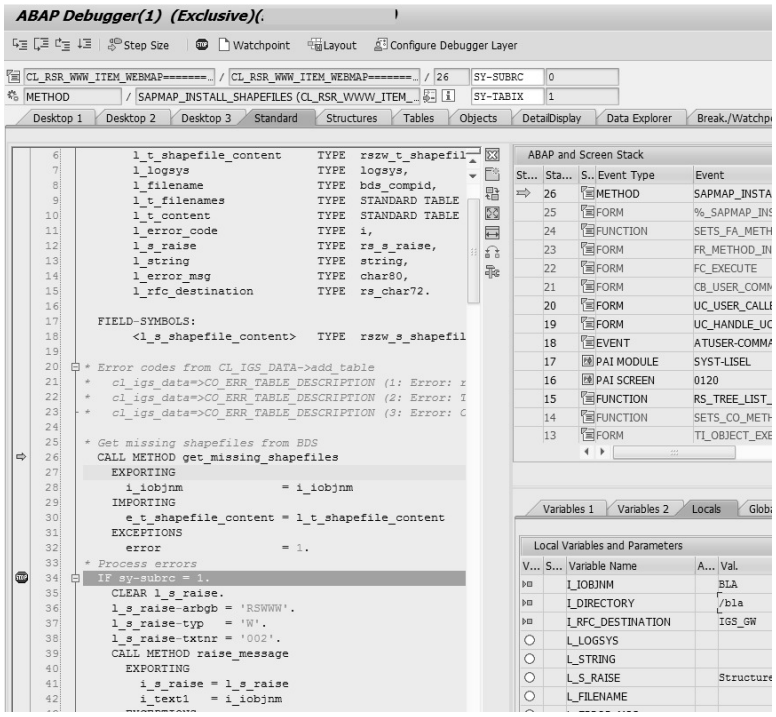

Fig. 7. ABAP Debugger

Le debugger SAP est très complet, et permet d'effectuer à peu près tout ce qu'on veut sur le programme : breakpoint, wachtpoint, modification de données, mais aussi exécution de scripts à certains moments, etc. Après avoir forcé la première vérification pour faire croire qu'il y a des shapefiles manquants, il est possible d'exécuter pas à pas la méthode pour l'étudier comme avec n'importe quel debugger. Il est également possible de modifier manuellement le contenu de la table envoyée par la méthode. En utilisant le debugger, nous arrivons effectivement à créer trois fichiers sur le serveur SAP via le service IGS. Ces fichiers sont stockés par défaut dans `/usr/sap/SID/D00/igs/data`. Reprenant l'architecture de test (voir figure 3), l'étude des paquets liés à cette fonction est devenue possible. Aussi en utilisant le nouveau module PySAP [1], il est possible d'automatiser ces tests.

5.3 Vulnérabilité découverte

Cette étude a mis au jour plusieurs vulnérabilités sur cette fonction. Les attaques suivantes sont toutes réalisables à distance et sans être authentifié. En revanche elles se font en aveugle, car nous n'avons pas la possibilité d'obtenir le code retour d'une requête et donc de savoir si elle a bien abouti ou pas.

Les attributs des fichiers shapefiles sont envoyées au service IGS sous forme d'une table. Les entrées de cette table ne sont pas vérifiées et un nom de fichier d'une longueur de 1266 caractères, écrase le contenu de certains registres, provoquant un crash du service SAP IGS, listing 2.

```
# python igs_adminstall_dos.py -d sapserver -p 40000 -v
[*] Testing ADM:INSTALL Untrusted Pointer Dereference on sapserver
    :40000
[*] Sending payload...
DEBUG:pysap.sapni:To send 4673 bytes
[*] File sent.
#

---

(gdb) c
Continuing.

Thread 2 "igspw_mt" received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7fccd6d3f700 (LWP 8636)]
__GI___libc_free (mem=0x4242424242424242) at malloc.c:2966
2966    malloc.c: No such file or directory.
(gdb)
```

Listing 2. PoC déni de service

De plus, le type de fichier n'est pas vérifié et il est possible d'écrire n'importe quel fichier dans le répertoire de donnée de l'IGS. Il est possible d'accéder à ces fichiers par la suite via l'interface HTTP du service, listing 3.

```
# curl http://sapserver:40080/output/sstic.txt
<HTML><HEAD><TITLE>SAP Internet Graphics Server</TITLE></HEAD><BODY
    ><H2><B>500 Internal Server Error</B></H2><BR><HR><BR>File not
    found<BR><BR><HR></BODY></HTML>
# echo "SSTIC_2018!" > sstic.txt
# python igs_adminstall_upload.py -d sapserver -p 40000 -i sstic.txt
[*] Testing ADM:INSTALL arbitrary file upload on sapserver:40000
[*] Uploading... output/sstic.txt
[*] File sent.
# curl http://sapserver:40080/output/sstic.txt
SSTIC_2018!
```

Listing 3. PoC téléversement de fichier

6 Conclusion

Spécifique n'est pas un synonyme de sécurisé. Cette étude montre que même SAP ne déroge pas à la règle. Dans ce cas il est important de mettre à jour le composant IGS [3, 7–9], mais aussi de filtrer les accès au service, voire de le désactiver [6].

Par ailleurs, via cet article, j'espère avoir montré que la recherche en sécurité SAP n'est pas si compliquée et qu'elle peut concerner plusieurs aspect de la sécurité (réseau, reverse, etc.).

Références

1. Martin Gallo. PySAP. https://github.com/CoreSecurity/pysap.
2. Martin Gallo. SAP-Dissection. https://github.com/CoreSecurity/SAP-Dissection-plug-in-for-Wireshark.
3. SAP AG. Open Source Software Security Vulnerabilities in SAP Internet Graphics Server (IGS). https://launchpad.support.sap.com/#/notes/2538829.
4. SAP AG. SAP Facts. https://www.sap.com/corporate/en/company.fast-facts.html#fast-facts.
5. SAP AG. SAP IGS Online documentation. https://help.sap.com/viewer/3348e831f4024f2db0251e9daa08b783/7.5.10/en-US/4e193ea5b5c617e2e10000000a42189b.html.
6. SAP AG. SAP Security Notes. https://support.sap.com/securitynotes.
7. SAP AG. Security Note. https://launchpad.support.sap.com/#/notes/2616599.
8. SAP AG. Security Note. https://launchpad.support.sap.com/#/notes/2615635.
9. SAP AG. Security vulnerabilities in SAP Internet Graphics Server (IGS). https://launchpad.support.sap.com/#/notes/2525222.
10. Yvan Genuer. SAP IGS : The 'vulnerable' forgotten component. https://troopers.de/troopers18/agenda/tr18-the-vulnerable-forgotten-component/, 2018.