



Shameful secrets of proprietary protocols



OWASP

The Open Web Application Security Project

Sławomir Jasek

Jakub Kałużny

Who are we



OWASP

The Open Web Application Security Project



Sławomir Jasek



Jakub Kałużny

- Pentesters @ SecuRing
- Security assessments of applications, networks, systems...



OWASP

The Open Web Application Security Project

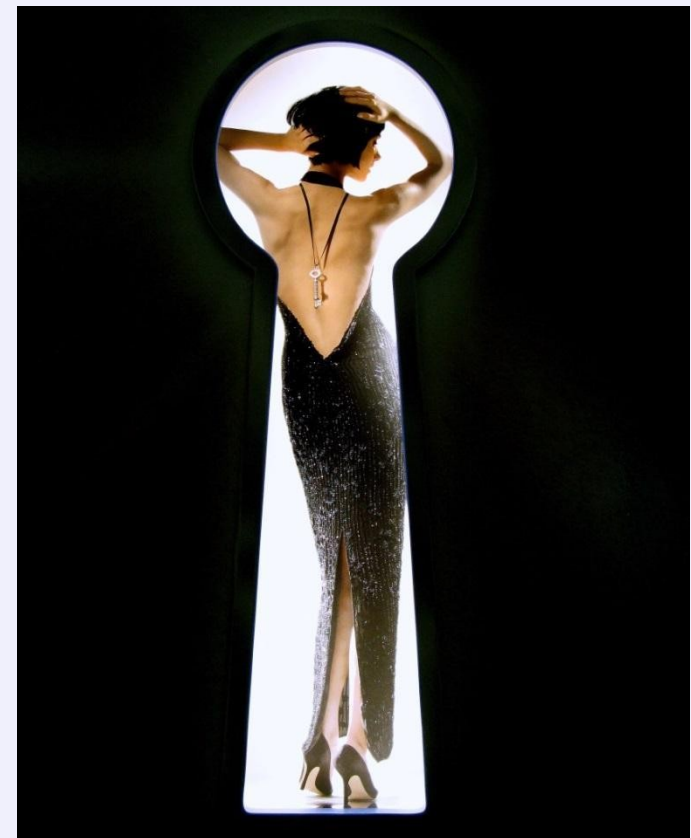
- Case studies – proprietary protocols
 - Home automation
 - Pull printing #1
 - Remote desktop
 - Pull printing #2
 - Trading
- Cheatsheet for architects & developers
- How to hack it



OWASP

The Open Web Application Security Project

- A pentester will encounter one
- Don't have the protocol specs nor tools to attack it
- How to hack it?
 - decompile the client?
 - search for some tools?
 - watch the raw packets?
- Let's try!





- „Plug the device, configure your router for port forwarding (and dynamic dns if necessary), set password.”
- Proprietary TCP protocol, direct connection from Internet to device, password protected access

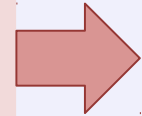




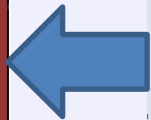
OWASP

The Open Web Application Security Project

```
ab 55 41 00 15 39 64 64 34 65 34 36 31 32 36   JA .9dd 4e46126  
02 01 00 00 a9 39 64 64 34 65 34 36 31 32 36   ....9dd 4e46126  
aa 55 41 00 14 39 64 64 34 65 34 36 31 32 36   JA .9dd 4e46126
```



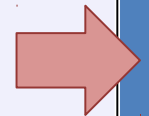
```
aa 53 41 02 01 01 f0 fl fl fl fl 00 be fl fl 00 SA.....  
c4 00 e1 fl fl fl fl fl fl fl fl fl fl fl fl .....  
fl fl fl 00 64 00 00 00 01 00 f0 f0 0a fl 00 02 ...d...  
0f0fe7
```



CLI
EN
T

SER
VE
R

```
ab 55 41 00 15 39 64 64 34 65 34 36 31 32 36   JA .9dd 4e46126  
0c 02 00 00 a4 39 64 64 34 65 34 36 31 32 36   ....9dd 4e46126  
aa 55 41 00 14 39 64 64 34 65 34 36 31 32 36   JA .9dd 4e46126
```



And what if we change the password?



OWASP

The Open Web Application Security Project

Password 1:

```
00000000 aa 55 41 00 14 39 32 65 62 35 66 66 65 65 36 .UA..92e b5ffee6
00000000 aa 53 41 02 01 01 f0 f1 f1 f1 f1 00 a1 f1 f1 00 .SA.....
00000010 92 00 dd f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 .....
00000020 f1 f1 f1 00 78 00 02 00 00 00 f0 f0 07 a3 00 02 ....x...
00000030 0f 0f d2 ...
```

Password 2

```
00000000 aa 55 41 00 14 34 61 38 61 30 38 66 30 39 64 .UA..4a8 a08f09d
00000000 aa 53 41 02 01 01 f0 f1 f1 f1 f1 00 a1 f1 f1 00 .SA.....
00000010 93 00 dd f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 .....
00000020 f1 f1 f1 00 78 00 02 00 00 00 f0 f0 07 a3 00 02 ....x...
00000030 0f 0f d3 ...
```

Password 3

```
00000000 aa 55 41 00 14 30 63 63 31 37 35 62 39 63 30 .UA..0cc 175b9c0
00000000 aa 53 41 02 01 01 f0 f1 f1 f1 f1 00 a1 f1 f1 00 .SA.....
00000010 92 00 dd f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 f1 .....
00000020 f1 f1 f1 00 78 00 02 00 00 00 f0 f0 07 a3 00 02 ....x...
00000030 0f 0f d2 ...
```



Internal command (5 bytes)

MD5(password) - first 10 bytes

00000000	aa 55 41 00 14	39 32 65 62 35 66 66 65 65 36	.UA..	92e b5fee6
00000000	aa 53 41 02 01 01 f0 f1	f1 f1 f1 00 a1 f1 f1 00	.SA.....
00000010	92 00 dd f1 f1 f1 f1 f1	f1 f1 f1 f1 f1 f1 f1 f1
00000020	f1 f1 f1 00 78 00 02 00	00 00 f0 f0 07 a3 00 02x...
00000030	0f 0f d2	

Status returned by the appliance (sensors, settings, etc).



- Sniffing
- MITM
- Connect directly to the appliance - sniffed hash is enough
- Recommendation: SSL!



OWASP

The Open Web Application Security Project

- Vendor: OK, we have added SSL support!

```
sslcontext = SSLContext.getInstance("TLS");
```

```
atrustmanager = new TrustManager[1];
```

```
atrustmanager[0] = new EasyX509TrustManager(null);
```

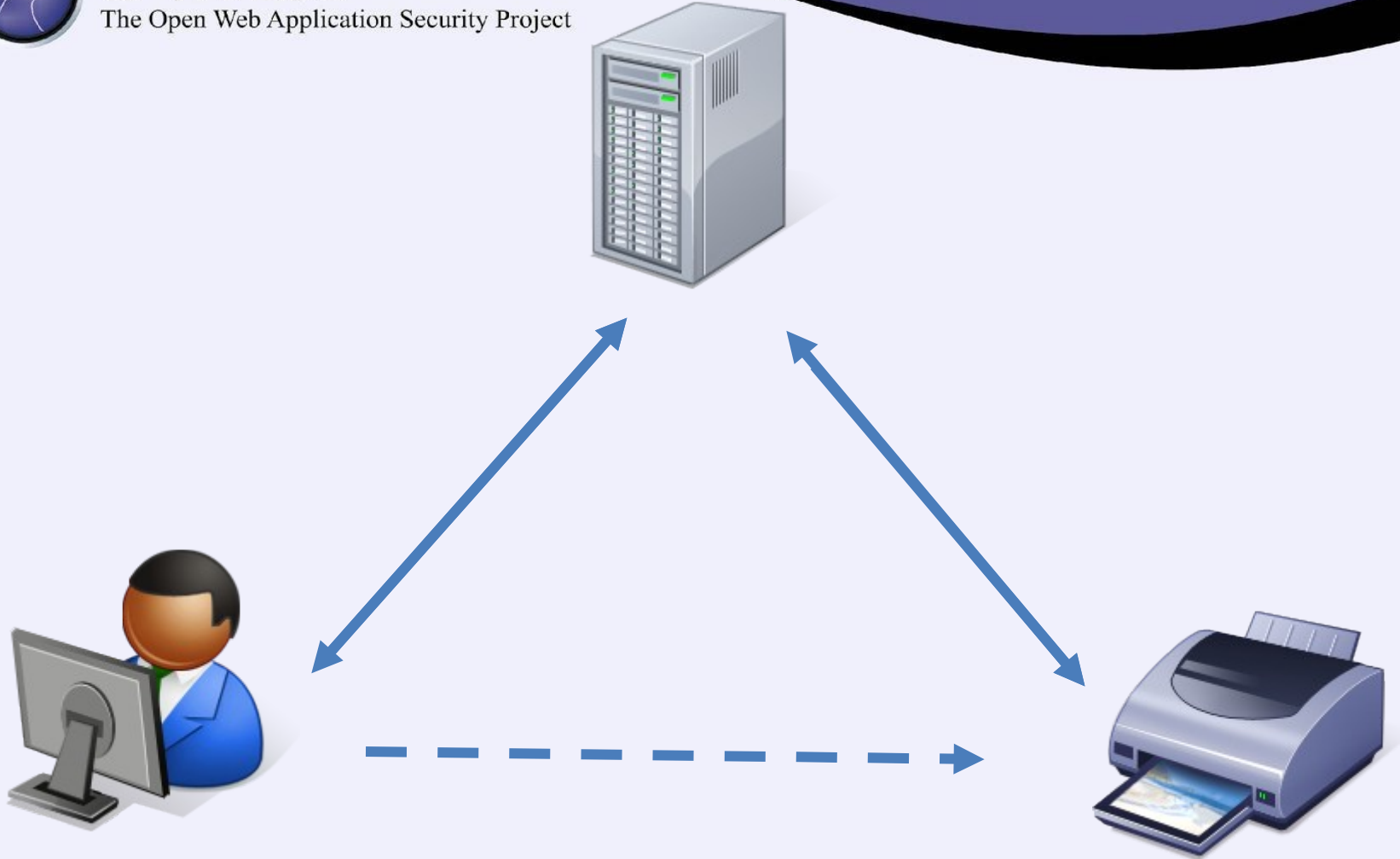
```
sslcontext.init(null, atrustmanager, null);
```

- Empty TrustManager – accepts all certificates



OWASP

The Open Web Application Security Project



Why hack pull printing?



OWASP

The Open Web Application Security Project

- Widely used
- Confidential data
- Getting popular



OWASP

The Open Web Application Security Project

sniffing

print queues

accountability

users' data

Attack vectors

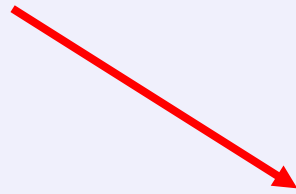


OWASP

The Open Web Application Security Project



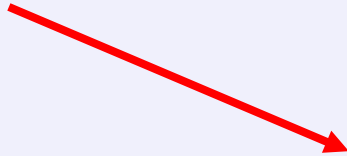
Other users' data



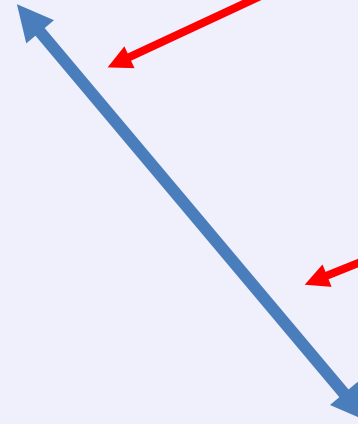
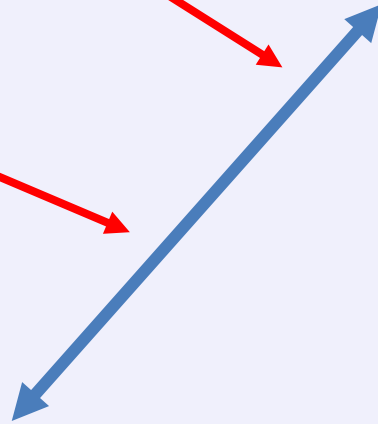
Sniffing, MITM



User/admin interface vulnerabilities



Access to other print queues



Authorization bypass





OWASP

The Open Web Application Security Project

“**Secure print release** (...) can integrate card-swipe user authentication at devices (...) ensuring jobs are **only** printed when the collecting user is present.”

Pull Printing #1 - binary protocol



OWASP

The Open Web Application Security Project

SER
VER



PRI
NTE
R



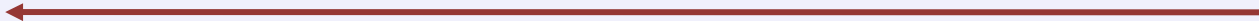
OWASP

The Open Web Application Security Project

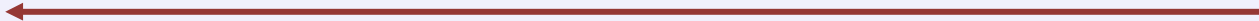
SERVER

PRINTER

Release print queue for user "guest-xyz" print queue



Charge user "guest-xyz" for copying 100 pages



```

65 64 54 53 00 S_restrictedTS.
70 79 54 53 00 canColorCopy S.
6c 69 65 72 44 .costMultiplierD
63 61 6e 43 68 ?..... S..canCh
72 6f 6d 4c 69 argeShar edFromLi
72 69 6e 74 4a stFS..he ldPrintJ
00 53 00 19 68 obCountI ....S..h
63 63 6f 75 6e asAdvanc edAccoun
tOptions Fzz
59 63 65 41 c..m.%ex tDeviceA
53 65 54 72 PI.begin DeviceTr
5d 4e 39 42 ansaction sS..mN9B
75 65 73 74 KS..1004 S..quest
-xyzS..z
75 73 53 00 07 T..MS..s tatsS..
76 61 69 6c 61 SUCCESSSS ..availa
ff d7 0a 3d 70 bleCredi tD?...=p
65 44 3f ff d7 ..S..bal anceD?..
74 75 73 4d 65 .=p..S.. statusMe
74 72 61 6e 73 ssageS.. S..trans
5a 70 44 35 30 actionId S..ZpD50
zz

```

canColorCopy

asAdvancedAccountOptions Fzz

PI.begin DeviceTransaction S..mN9BKS..1004 S..quest-xyzS..z

User permissions

beginDeviceTransaction
(...) guest-xyz

```

59 63 65 41 c..m.%ex tDeviceA
43 6f 70 69 PI.calcu lateCopi
30 05 6d 4e erPageCo stsS..mN
39 67 75 65 9BKS..10 04S..gue
34 46 46 7a st-xyzVV S..A4FFz

```



OWASP

The Open Web Application Security Project

sniffing

print queues

accountability

users' data



OWASP

The Open Web Application Security Project

- Gave access to KB and support service
- And all versions of software
- Responded in few hours and patched in few days
- Was happy to be pentested



OWASP

The Open Web Application Security Project

- X-win „on steroids” (encryption, compression, access control...)
- Mainframe access for critical business operations
- „More than 100,000 users around the world”
- „Prevents unauthorized eavesdropping”

FIPS 140-2 Validated

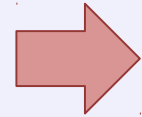
End-to-end data encryption”





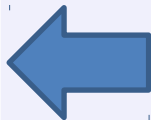
00000000 01 01 00 00

....



00000000 01 00 00 00

....



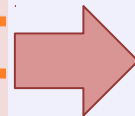
00000004 16 03 00 00 6d 01 00 00 69 03 00 52 8d e8 02 cf ...m ...i.R....

CLI
EN
T

00000004 11 01 30 0d 08 03 f1 00 00 00 00 00 00 00 00 ..0.....

00000014 00 ff ff 7f 00 00 01 ac 3d 08 08 68 69 6a 61 63 =..hijac

00000024 6b 65 64 0a 30 35 31 45 31 45 31 41 32 36 00 01 ked.051E 1E1A26 .



00000054 00 12 00 11 00 0a 00 09 00 08 00 07 00 06 00 05

SER
VE
R

LOGIN

ENCODED
PASSWORD



Password



OWASP

The Open Web Application Security Project

54657374696e67506173TestingPassword1234TestingPassword

XOR

1c101e190000032080117572c1d095c475d5d3704071d060014702d1a1e1e1b1700

=

48756d61696e676769726420436f6d6d756e69636174696f6e73204c696d69746564
[redacted] Communications Limited



default configuration

CLIENTHELLO!

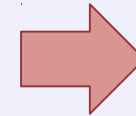
cipher suites:

SSL_DHE_RSA_WITH_AES_256_CBC_SHA

SSL_DHE_DSS_WITH_AES_256_CBC_SHA

SSL_RSA_WITH_AES_256_CBC_SHA

(...)



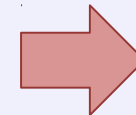
SERVERHELLO!

I don't have any certificate!

cipherSuite: SSL_DH_anon_WITH_AES_256_CBC_SHA



OK, no problem! You have to be the right server if you say so, don't you?



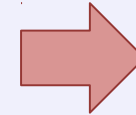
CLI
EN
T

SER
VE
R



certificates configured

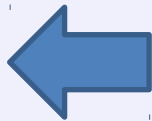
CLIENTHELLO!



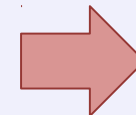
SERVERHELLO!

I don't have any certificate!

cipherSuite: SSL_DH_anon_WITH_AES_256_CBC_SHA



I have your certificate, but since you don't offer it any more, I won't check it. OK, let's connect!



CLI
EN
T

SER
VE
R



OWASP

The Open Web Application Security Project

- „We don't know PGP, use zip with our CEO's name as password”
- Do not plan to solve the issues (?)
- `> /dev/null 2>&1`
- Full disclosure!
- ... and a few weeks later the mysterious shut down of our beloved ;)



OWASP

The Open Web Application Security Project

“is a modern printing solution that **safeguards document confidentiality** and unauthorized access to print, scan, copy and e-mail functions. Its user-authentication **provides air-tight security** on your shared MFPs that function as personal printers.”

Vendor ensures



OWASP

The Open Web Application Security Project

„Documents are delivered **only** into the right hands”

„Information is kept **confidential**. **No risk** of being left unattended at the printer”

„Document collection is **safe anytime and anywhere** — no “print and sprint”. ”

„Integration with other enterprise applications and workflows **is kept secure** through single sign-on”



OWASP

The Open Web Application Security Project

First look on communication:

- TCP, 2 ports
- No cleartext, no SSL
- Seems to follow some scheme...



OWASP

The Open Web Application Security Project

- Hardcoded RSA certificate in printer embedded software
- No trust store
- AES-128 ECB used for traffic encryption
- Same protocol in admin interface



OWASP

The Open Web Application Security Project

sniffing

print queues

accountability

users' data



OWASP

The Open Web Application Security Project

“(...) system has been deployed at many high security customers and **has passed internal audits.**”



OWASP

The Open Web Application Security Project

- An online application for instant financial operations
- A proprietary, binary protocol, designed in order to minimise delays
- TCP in SSL tunnel





OWASP

The Open Web Application Security Project

Follow TCP Stream

Stream Content

```
000001C6 00 00 00 0e 00 5c 00 00 00 00 00 01 01 00 00 .....\. . . . .
000001D6 00 39 .g
00000000 00 00 00 22 00 00 00 39 00 06 00 00 00 00 00 00 ..."...9 . . . . .
00000010 01 01 00 5c 00 00 00 00 00 00 01 01 00 5c 00 00 ...\. . . . .
00000020 00 00 00 00 00 96
00000026 00 00 02 77 00 00 00 48 00 13 00 00 01 00 00 4a ...w...H . . . . .J
00000036 00 00 00 03 02 5b 00 76 00 00 00 00 00 73 02 51 .....[.v . . . . .s.Q
00000046 3c 74 61 62 6c 65 20 6e 61 6d 65 3d 22 43 75 73 <table n ame="Cus
00000056 74 6f 6d 20 52 65 70 6f 72 74 3a 20 41 63 74 69 tom Repo rt: Acti
00000066 76 65 20 4f 72 64 65 72 73 20 52 65 70 6f 72 74 ve Order s Report
00000076 22 20 66 65 74 63 68 53 69 7a 65 3d 22 30 22 3e " fetchS ize="0">
00000086 3c 74 68 3e 20 3c 74 64 20 74 79 70 65 3d 22 53 <th> <td type="S
00000096 54 52 49 4e 47 22 3e 6c 6f 67 69 6e 3c 2f 74 64 TRING">l ogin</td
000000A6 3e 3c 74 64 20 74 79 70 65 3d 22 53 54 52 49 4e ><td typ e="STRIN
000000B6 47 22 3e 61 63 63 6f 75 6e 74 3c 2f 74 64 3e 3c G">accou nt</td><
000000C6 74 64 20 74 79 70 65 3d 22 53 54 52 49 4e 47 22 td type= "STRING"
000000D6 3e 67 72 6f 75 70 6e 61 6d 65 3c 2f 74 64 3e 3c >groupna me</td><
000000E6 74 64 20 74 79 70 65 3d 22 49 4e 54 45 47 45 52 td type= "INTEGER
000000F6 22 3e 6f 72 64 65 72 69 64 3c 2f 74 64 3e 3c 74 ">orderi d</td><t
00000106 64 20 74 79 70 65 3d 22 44 41 54 45 22 3e 63 72 d type=" DATE">cr
00000116 65 61 74 65 64 61 74 3c 2f 74 64 3e 3c 74 64 20 eatedat< /td><td
00000126 74 79 70 65 3d 22 53 54 52 49 4e 47 22 3e 62 6f type="ST RING">bo
00000136 75 6e 64 74 6f 6f 72 64 65 72 69 64 3c 2f 74 64 undtoord erid</td
00000146 3e 3c 74 64 20 74 79 70 65 3d 22 53 54 52 49 4e ><td typ e="STRIN
00000156 47 22 3e 69 6e 73 74 72 75 6d 65 6e 74 6e 61 6d G">instr umentnam
00000166 65 3c 2f 74 64 3e 3c 74 64 20 74 79 70 65 3d 22 e</td><t d type="
00000176 53 54 52 49 4e 47 22 3e 6f 70 65 6e 63 6c 6f 73 STRING"> openclos
00000186 65 62 75 79 73 65 6c 6c 3c 2f 74 64 3e 3c 74 64 ebuysell </td><td
00000196 20 74 79 70 65 3d 22 53 54 52 49 4e 47 22 3e 6f type="S TRING">o
000001A6 72 64 65 72 74 79 70 65 3c 2f 74 64 3e 3c 74 64 rdertype </td><td
000001B6 20 74 79 70 65 3d 22 4e 55 4d 42 45 52 22 3e 70 tvne="N UMBER">n
```

Entire conversation (1145 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

That's interesting!



OWASP

The Open Web Application Security Project

Follow TCP Stream

Stream Content

```
00000000 00 00 00 44 00 00 01 91 00 13 00 00 00 01 00 16 ...D....
00000010 00 0d 43 6c 69 65 6e 74 53 72 76 2e 6a 77 73 00 ..Client Srv.jws.
00000020 4a 00 00 00 0b 00 17 00 e7 00 06 6d 65 74 68 6f J..... ..metho
00000030 64 00 e6 00 09 69 73 43 6c 75 73 74 65 72 00 5c d....isC luster.\
00000040 00 00 00 00 00 00 00 01
00000000 00 00 01 d2 00 00 01 92 00 13 00 00 00 01 00 4a .....J
00000010 00 00 00 03 01 b6 00 76 00 00 00 00 00 73 01 ac .....v .....S..
00000020 3c 3f 78 6d 6c 20 76 65 72 73 69 6f 6e 3d 22 31 <?xml ve rsion="1
00000030 2e 30 22 20 65 6e 63 6f 64 69 6e 67 3d 22 55 54 .0" enco ding="UT
00000040 46 2d 38 22 3f 3e 3c 73 6f 61 70 65 6e 76 3a 45 F-8"?><s oapenv:E
00000050 6e 76 65 6c 6f 70 65 20 78 6d 6c 6e 73 3a 73 6f nvelope xmlns:so
00000060 61 70 65 6e 76 3d 22 68 74 74 70 3a 2f 2f 73 63 apenv="h ttp://sc
00000070 68 65 6d 61 73 2e 78 6d 6c 73 6f 61 70 2e 6f 72 hemas.xml soap.or
00000080 67 2f 73 6f 61 70 2f 65 6e 76 65 6c 6f 70 65 2f g/soap/e nvelope/
00000090 22 20 78 6d 6c 6e 73 3a 78 73 64 3d 22 68 74 74 " xmlns: xsd="htt
000000A0 70 3a 2f 2f 77 77 77 2e 77 33 2e 6f 72 67 2f 32 p://www. w3.org/2
000000B0 30 30 31 2f 58 4d 4c 53 63 68 65 6d 61 22 20 78 001/XMLS chema" x
000000C0 6d 6c 6e 73 3a 78 73 69 3d 22 68 74 74 70 3a 2f mlns:xsi ="http:/
000000D0 2f 77 77 77 2e 77 33 2e 6f 72 67 2f 32 30 30 31 /www.w3. org/2001
000000E0 2f 58 4d 4c 53 63 68 65 6d 61 2d 69 6e 73 74 61 /XMLSche ma-insta
000000F0 6e 63 65 22 3e 3c 73 6f 61 70 65 6e 76 3a 42 6f nce"><so apenv:Bo
00000100 64 79 3e 3c 69 73 43 6c 75 73 74 65 72 52 65 73 dy><isCl usterRes
00000110 70 6f 6e 73 65 20 73 6f 61 70 65 6e 76 3a 65 6e ponse so apenv:en
00000120 63 6f 64 69 6e 67 53 74 79 6c 65 3d 22 68 74 74 codingSt yle="htt
00000130 70 3a 2f 2f 73 63 68 65 6d 61 73 2e 78 6d 6c 73 p://sche mas.xmls
00000140 6f 61 70 2e 6f 72 67 2f 73 6f 61 70 2f 65 6e 63 oap.org/ soap/enc
00000150 6f 64 69 6e 67 2f 67 2f 73 6f 61 70 2f 65 6e 63 oap.org/ soap/enc
```

Entire conversation (631 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

That's interesting!



OWASP

The Open Web Application Security Project

Follow TCP Stream

Stream Content

```
....D.....
ClientSrv.jws.J.....method....isCluster.
\.....J.....v.....s...<?xml version="1.0"
encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/
XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><soapenv:Body><isClusterResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/"><isClusterReturn xsi:type="xsd:string">false</
isClusterReturn></isClusterResponse></soapenv:Body></
soapenv:Envelope>.\.....6.s.$Connection was interrupted by
client...8.\.....8.s..Error.....\.....
```

Entire conversation (631 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

And what if we...



OWASP

The Open Web Application Security Project

Follow TCP Stream

Stream Content

```
...W.\.....?.....  
AdminService.J.....method....AdminService.J.....aaa....bbb...@.....J.....  
$.v.....s...<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://  
schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://  
www.w3.org/2001/XMLSchema-  
instance"><soapenv:Body><soapenv:Fault><faultcode>soapenv:Server.userException</  
faultcode><faultstring>java.lang.Exception: Unable to process the message -was it a valid WSDO  
descriptor?</faultstring><detail><ns1:stackTrace xmlns:ns1="http://xml.apache.org/  
axis/">java.lang.Exception: Unable to process the message -was it a valid WSDO descriptor?  
.at org.apache.axis.utils.Admin.process(Admin.java:163)  
.at org.apache.axis.utils.Admin.AdminService(Admin.java:65)  
.at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
.at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)  
.at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)  
.at java.lang.reflect.Method.invoke(Method.java:597)  
.at org.apache.axis.providers.java.MsgProvider.processMessage(MsgProvider.java:126)
```

Entire conversation (2975 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

And how about...



OWASP

The Open Web Application Security Project

Follow TCP Stream

Stream Content

```
...D.  
\\.....?.....ClientSrv.jws.J.....method.....RegisterUser.....  
J.....e.v.....s.[<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope  
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://  
www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"><soapenv:Body><registerUserResponse soapenv:encodingStyle="http://  
schemas.xmlsoap.org/soap/encoding/"><registerUserReturn  
xsi:type="xsd:string">&lt;error code="266" &gt;Incorrect login&lt;/  
error&gt;</registerUserReturn></registerUserResponse></  
soapenv:Body><soapenv:Envelope>.
```

Entire conversation (717 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

? Help Filter Out This Stream Close



```
<soapenv:Body>
```

```
  <registerUserResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
```

```
    <registerUserReturn xsi:type="xsd:string">
```

```
      &lt;error code="266" &gt;Incorrect login&lt;/error&gt;
```

```
    </registerUserReturn>
```

```
  </registerUserResponse>
```

```
</soapenv:Body>
```

- Incorrect password
- Incorrect first name
- Group with name null doesn't exist
- Group with name admin doesn't exist
- Group with name Administrator doesn't exist
- And how about „root“?

Game Over



OWASP

The Open Web Application Security Project

```
< soapenv:Body>  
  < registerUserResponse soapenv:encodingStyle= "http://schemas.xmlsoap.org/soap/encoding/">  
    < registerUserReturn xsi:type= "xsd:string">  
      User was registered successfully with id= 5392745  
    </registerUserReturn>  
  </registerUserResponse>  
</soapenv:Body>
```

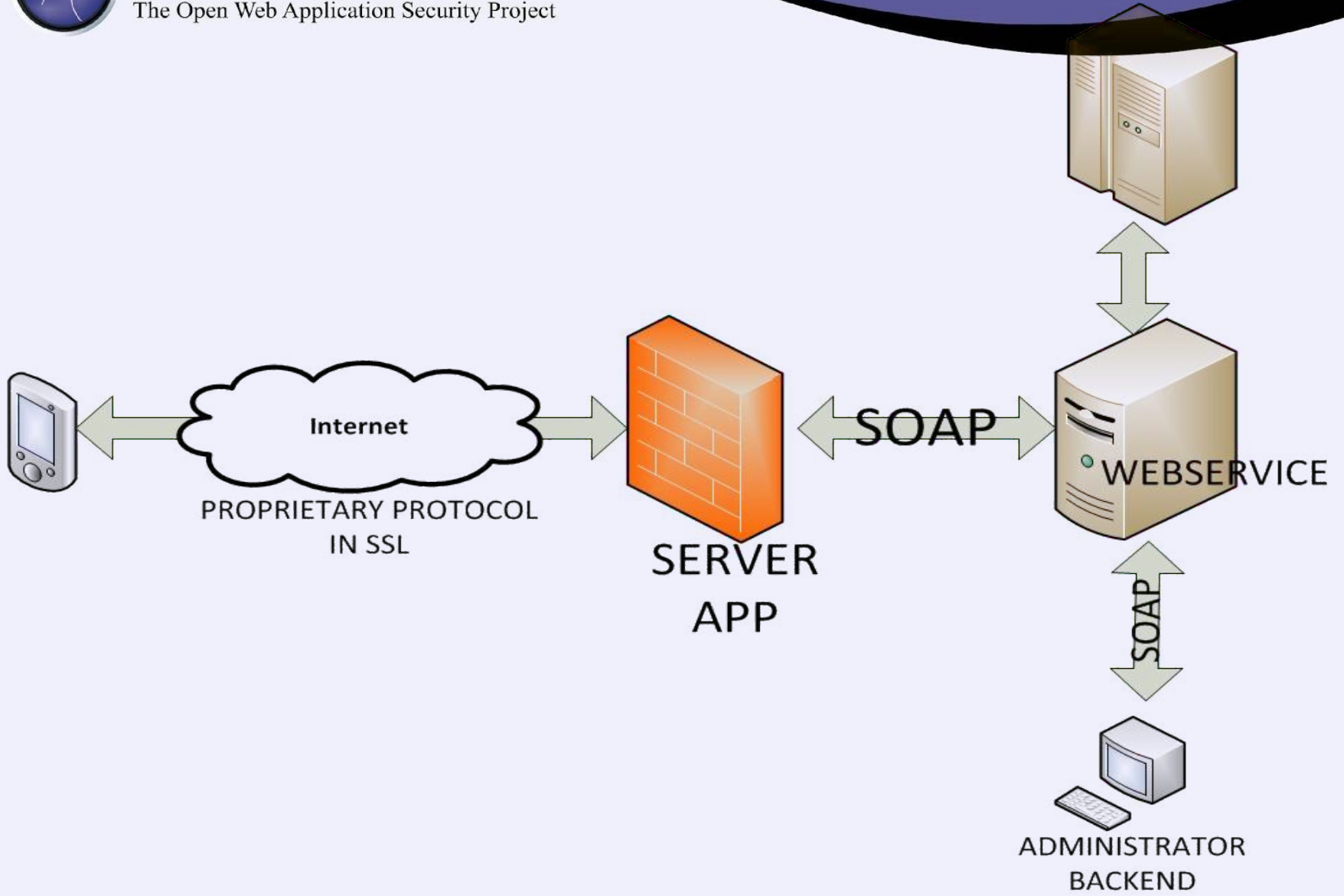


So now we can manage all the other accounts and spend their money!



OWASP

The Open Web Application Security Project





While deploying a proprietary solution:

- Get it pentested
- Verify vendor claims
- Ask the vendor for secure development lifecycle, procedures of addressing vulnerabilities, previous bugs



OWASP

The Open Web Application Security Project

- Protocol is NOT secure by its secrecy
- Proper encryption. Use known standards, implement them with care.
- Input validation, access control, many layers of security, least privilege principle...
- Beware backwards compatibility



Decompile client?

Inject code?

Search for the specs?

Use some tools?

Watch the packets?



Look for the fine manual



OWASP

The Open Web Application Security Project

- There may be unofficial client, or e.g. wireshark plugin
- Ask for the docs 😊
- Search for them
 - Yes, we have found internal protocol specification by google hacking!



- Sometimes easy – e.g. not obfuscated Android application:

```
byte abyte3 [] = pass.getBytes();  
byte abyte4 [] = MessageDigest.getInstance("MD5").digest(abyte3);  
String s1 = "";  
for(int j= 0; j< 10; j+ )  
s1 = (new  StringBuilder()).append(s1).append(toHexString(abyte4).charAt(j)).toString();  
System.arraycopy(s1.getBytes(),0,abyte1,5,10);
```

- Sometimes really hard & time consuming.
- May be fun, but often leads astray



- Various tools to analyze proprietary protocols
 - time consuming, usually do not work
- Raw, just try to spot some scheme
 - of course with a little help of your friends: wireshark, tcpdump, ssldump etc.
- Your favourite scripting language

Q&A?



OWASP

The Open Web Application Security Project

slawomir.jasek@securing.pl

jakub.kaluzny@securing.pl

www.securing.pl