

UNIVERSIDADE DO MINHO  
CRIPTOGRAFIA E SEGURANÇA DA INFORMAÇÃO  
DEPARTAMENTO DE INFORMÁTICA

ENGENHARIA DE SEGURANÇA

---

## TP2 - Aula 3

---

*Grupo 7*

Carlos Pinto Pedrosa A77320

José Francisco Gonçalves Petejo e Igreja Matos A77688

25 de Fevereiro de 2019

# Conteúdo

<b>1</b>	<b>Blind Signatures</b>	<b>2</b>
1.1	Alterações do Código das diferentes Classes . . . . .	2
1.1.1	initSigner-app.py . . . . .	2
1.1.2	generateBlindSignature-app.py . . . . .	3
1.1.3	generateBlindData-app.py . . . . .	4
1.1.4	unblindSignature-app.py . . . . .	5
1.1.5	verifySignature-app.py . . . . .	6
<b>2</b>	<b>Protocolo SSL/TLS</b>	<b>9</b>
2.1	SSL Server Test Câmaras Municipais Portuguesas . . . . .	9
2.1.1	Braga . . . . .	9
2.1.2	Porto . . . . .	10
2.1.3	Guimarães . . . . .	10
2.2	Segurança do site da Câmara Municipal de Braga . . . . .	11
2.3	POODLE (TLS) . . . . .	12
<b>3</b>	<b>Protocolo SSH</b>	<b>13</b>
3.1	Testes ssh de empresas comerciais de Madrid . . . . .	13
3.1.1	Vodafone Spain - 77.226.253.213 . . . . .	14
3.1.2	Axarnet Communications - 91.141.211.210 . . . . .	15
3.2	Software e Versões utilizadas pelos servidores ssh . . . . .	16
3.3	Vulnerabilidades . . . . .	16
3.3.1	Vulnerabilidade mais grave . . . . .	17
3.3.2	Gravidade das Vulnerabilidades . . . . .	18

# 1 Blind Signatures

Esta pergunta tem como objetivo a alteração do código *Python* de modo a responder ao enunciado. Assim, neste capítulo, irá-se descrever todas as alterações efetuadas para responder às perguntas.

## 1.1 Alterações do Código das diferentes Classes

### 1.1.1 initSigner-app.py

```
def printUsage():
    print("Usage: python initSigner-app.py")
    print("Usage: python initSigner-app.py -init")

def parseArgs():
    if (len(sys.argv) > 2):
        printUsage()
    if (len(sys.argv) == 2 and sys.argv[1] == "-init"):
        initComps()
    else:
        main()

def initComps():
    initComponents, pRDashComponents = eccblind.initSigner()
    iC = open("initcomponents.txt", "w")
    pRDC = open("pRDashComponents.txt", "w")
    iC.write initComponents
    pRDC.write pRDashComponents

def main():
    initComponents, pRDashComponents = eccblind.initSigner()
    print("Output")
    # print("Init components: %s" % initComponents)
    print("pRDashComponents: %s" % pRDashComponents)

if __name__ == "__main__":
    parseArgs()
```

Como se pode observar pelo código acima, as alterações necessárias foram as seguintes:

- Retirar o *print* do *initComponents* na *main()*;
- Alterar a *parseArgs()* para lidar com os argumentos;
- Criar a *initComponents()* para inicializar as variáveis e guardá-las em ficheiro.

### 1.1.2 generateBlindSignature-app.py

```
import sys
from eVotUM.Cripto import utils
from eVotUM.Cripto import eccblind

def printUsage():
    print("Usage: python generateBlindSignature-app.py " +
          "-key <chave privada> -bmsg <Blind message>")

def parseArgs():
    if (len(sys.argv) != 5):
        printUsage()
    elif ( sys.argv[1] == "-key" and sys.argv[3] == "-bmsg"):
        eccPrivateKeyPath = sys.argv[2]
        blindM = sys.argv[4]
        main(eccPrivateKeyPath, blindM)

def showResults(errorCode, blindSignature):
    print("Output")
    if (errorCode is None):
        print("Blind signature: %s" % blindSignature)
    elif (errorCode == 1):
        print("Error: it was not possible to retrieve the private key")
    elif (errorCode == 2):
        print("Error: init components are invalid")
    elif (errorCode == 3):
        print("Error: invalid blind message format")
```

```

def main(eccPrivateKeyPath, blindmessage):
    pemKey = utils.readFile(eccPrivateKeyPath)
    print("Input")
    passphrase = raw_input("Passphrase: ")
    initComponents = raw_input("Init components: ")
    blindM = blindmessage
    errorCode, blindSignature = eccblind.generateBlindSignature(pemKey,
                                                                passphrase, blindM, initComponents)
    showResults(errorCode, blindSignature)

if __name__ == "__main__":
    parseArgs()

```

Nesta situação, foi apenas necessário alterar a *parseArgs()* para lidar com os argumentos e alterar parcialmente a função *main()* para receber um novo argumento (a mensagem) e não o receber como *input* manual.

### 1.1.3 generateBlindData-app.py

```

import sys
from eVotUM.Cripto import eccblind

def printUsage():
    print("Usage: python generateBlindData-app.py " +
          "-msg <mensagem a assinar> -RDash <pRDashComponents>")

def parseArgs():
    if (len(sys.argv) != 5):
        printUsage()
    elif ( sys.argv[1] == "-msg" and sys.argv[3] == "-RDash"):
        data = sys.argv[2]
        rdash = sys.argv[4]
        main(data, rdash)

def showResults(errorCode, result):
    print("Output")
    if (errorCode is None):
        blindComponents, pRComponents, blindM = result

```

```

    print("Blind message: %s" % blindM)

    # print("Blind components: %s" % blindComponents)
    # print("pRComponents: %s" % pRComponents)

    requerente = open("requerente.txt", "w")
    requerente.write(blindComponents)
    requerente.write("abcd1234")
    requerente.write(pRComponents)

elif (errorCode == 1):
    print("Error: pRDash components are invalid")

def main(data, pRDashComponents):
    errorCode, result = eccblind.blindData(pRDashComponents, data)
    showResults(errorCode, result)

if __name__ == "__main__":
    parseArgs()

```

Novamente neste código, foi necessário:

- Alterar a função *parseArgs()* para lidar com os novos argumentos;
- Alterar a *main()* para lidar com mais um argumento e não receber input manual nenhum;
- Alterar a *showResults()* apenas imprimir a *Blind Message* e guardar em ficheiro as restantes variáveis (*blindComponents* & *pRComponents*).

#### 1.1.4 unblindSignature-app.py

```

import sys
from eVotUM.Cripto import eccblind

def printUsage():
    print("Usage: python unblindSignature-app.py " +
          "-s <Blind Signature> -RDash <pRDashComponents>")

```

```

def parseArgs():
    if (len(sys.argv) != 5):
        printUsage()
    elif (sys.argv[1] == "-s" and sys.argv[3] == "-RDash"):
        blindSignature = sys.argv[2]
        pRDashComponents = sys.argv[4]
        main(blindSignature, pRDashComponents)
    else:
        printUsage()

def showResults(errorCode, signature):
    print("Output")
    if (errorCode is None):
        print("Signature: %s" % signature)
    elif (errorCode == 1):
        print("Error: pRDash components are invalid")
    elif (errorCode == 2):
        print("Error: blind components are invalid")
    elif (errorCode == 3):
        print("Error: invalid blind signature format")

def main(blindSignature, pRDashComponents):
    print("Input")
    blindComponents = raw_input("Blind components: ")
    errorCode, signature = eccblind.unblindSignature(blindSignature, pRDashComponents, blindComponents)
    showResults(errorCode, signature)

if __name__ == "__main__":
    parseArgs()

```

Mais uma vez, foi necessário alterar a função *parseArgs()* para lidar com os novos argumentos e alterar a *main()* para só pedir o *blindComponents*.

### 1.1.5 verifySigature-app.py

```

import sys
from eVotUM.Cripto import eccblind
from eVotUM.Cripto import utils

```

```

def printUsage():
    print("Usage: python verifySignature-app.py " +
          "-cert <certificado do assinante> " +
          "-msg <mensagem original a assinar> " +
          "-sDash <Signature> -f <ficheiro do requerente>")

def parseArgs():
    if (len(sys.argv) != 9):
        printUsage()
    elif ( sys.argv[1] == "-cert" and sys.argv[3] == "-msg"
           and sys.argv[5] == "-sDash"
           and sys.argv[7] == "-f" ):
        eccPublicKeyPath = sys.argv[2]
        data = sys.argv[4]
        signature = sys.argv[6]

        s = open("requerente.txt", "r")
        txt = s.read()
        a = txt.find("abcd1234")

        blindComponents = txt[:a]
        pRComponents = txt[a+8:]

        main(eccPublicKeyPath, data, signature, blindComponents, pRComponents)
    else:
        printUsage()

def showResults(errorCode, validSignature):
    if (errorCode is None):
        if (validSignature):
            print("Valid signature")
        else:
            print("Invalid signature")
    elif (errorCode == 1):
        print("Error: it was not possible to retrieve the public key")
    elif (errorCode == 2):
        print("Error: pR components are invalid")

```



```

elif (errorCode == 3):
    print("Error: blind components are invalid")
elif (errorCode == 4):
    print("Error: invalid signature format")

def main(eccPublicKeyPath, data, signature, blindComponents, pRComponents):
    pemPublicKey = utils.readFile(eccPublicKeyPath)
    errorCode, validSignature = eccblind.verifySignature(pemPublicKey,
                                                         signature, blindComponents, pRComponents, data)
    showResults(errorCode, validSignature)

if __name__ == "__main__":
    parseArgs()

```

Por fim, para alterar a validação da assinatura, foi necessário:

- Alterar a função *parseArgs()* para lidar com os novos argumentos e retirar do ficheiro passado como argumento as variáveis pretendidas;
- Alterar a função *main()* para já não receber input manual;

Com estas novas alterações, o processo para validar uma assinatura torna-se ligeiramente diferente, como se pode verificar na figura abaixo.

```

user@CSI:~/Desktop/Aula3/Aula3_Modified/BlindSignatures$ python initSigner-app.py -init
user@CSI:~/Desktop/Aula3/Aula3_Modified/BlindSignatures$ python generateBlindData-app.py -msg "Engenharia de Segurança" -RDash "925d967ff9087410baddbaf7a3c72a782dc6c2a6bc94a285ae
ad7d17580084dd.d15432530e29c166705aabb3f3466ff4bc71d45ee65c5bfc5011d251cff355a0"
Input
Output
Blind message: fd61568013ed225e5031cd5636cf38455610ea22e1c1cf6af94bf2c3eece683
user@CSI:~/Desktop/Aula3/Aula3_Modified/BlindSignatures$ python generateBlindSignature-app.py -key key.pem -bmsg "fd61568013ed225e5031cd5636cf38455610ea22e1c1cf6af94bf2c3eece683
"
Input
Passphrase: 1234
Init components: 925d967ff9087410baddbaf7a3c72a782dc6c2a6bc94a285aead7d17580084dd.f2e3fc410ce2893255cc02853b0eed4324636dfe306de06f30dedf6e29df673f
Output
Blind signature: e03229a4c764576cadf7e7de1ca11cc6dfe87d44daf138d2f97c92baf607b21e3ea35fa339381bdc3f9591219e247a16b3588188d63463d8c6fdf874f8f9187d
user@CSI:~/Desktop/Aula3/Aula3_Modified/BlindSignatures$ python unblindSignature-app.py -s "e03229a4c764576cadf7e7de1ca11cc6dfe87d44daf138d2f97c92baf607b21e3ea35fa339381bdc3f9591
219e247a16b3588188d63463d8c6fdf874f8f9187d" -RDash "925d967ff9087410baddbaf7a3c72a782dc6c2a6bc94a285aead7d17580084dd.d15432530e29c166705aabb3f3466ff4bc71d45ee65c5bfc5011d251cff3
55a0"
Input
Blind components: 3b889a3e68b40df206fb4b38735c4d1a221b1cc9d99865aaee87542f3ab3ed8d.7bd9d6da7d65a7e0c72da2721071a6f66c0587b49aafa69c8f3b2c9b66892323
Output
Signature: 272ae90d7fc8853cdc4715ab167ae6dcfa4c6d6084d979dc109cae66abeae173
user@CSI:~/Desktop/Aula3/Aula3_Modified/BlindSignatures$ python verifySignature-app.py -cert key.crt -msg "Engenharia de Segurança" -sDash "272ae90d7fc8853cdc4715ab167ae6dcfa4c6d6
084d979dc109cae66abeae173" -f "requerente.txt"
Valid signature

```

Figura 1: Funcionamento do Software Modificado

## 2 Protocolo SSL/TLS

### 2.1 SSL Server Test Câmaras Municipais Portuguesas

#### 2.1.1 Braga

##### SSL Report: [www.cm-braga.pt](http://www.cm-braga.pt) (62.28.4.75)

Assessed on: Sat, 23 Feb 2019 22:38:11 UTC | [Hide](#) | [Clear cache](#)

[Scan Another »](#)

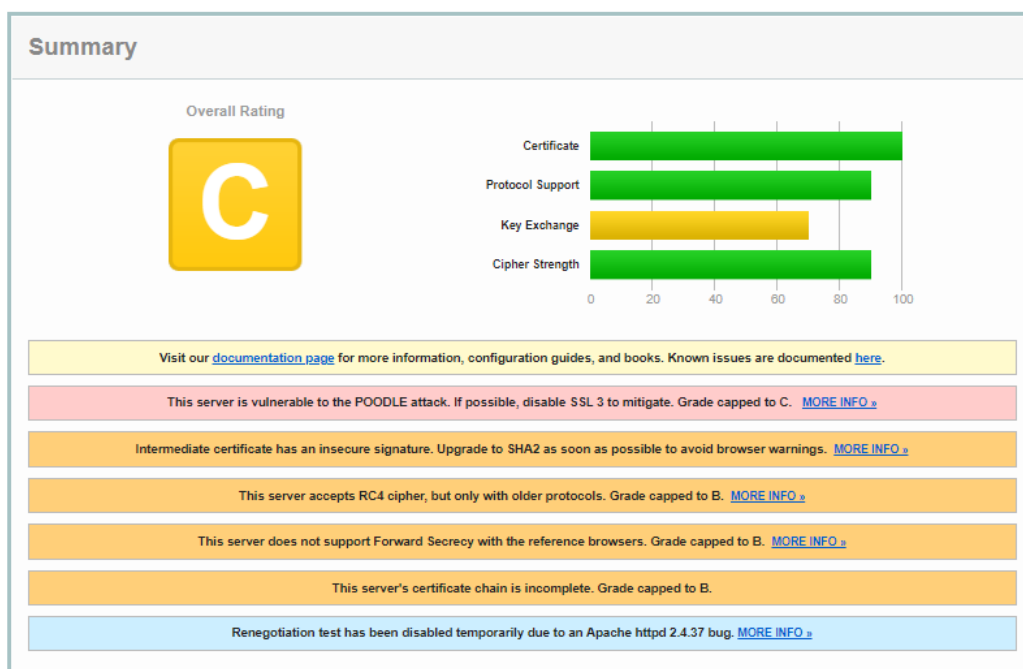


Figura 2: Resultado do SSL Server Test para cm-braga.pt

## 2.1.2 Porto

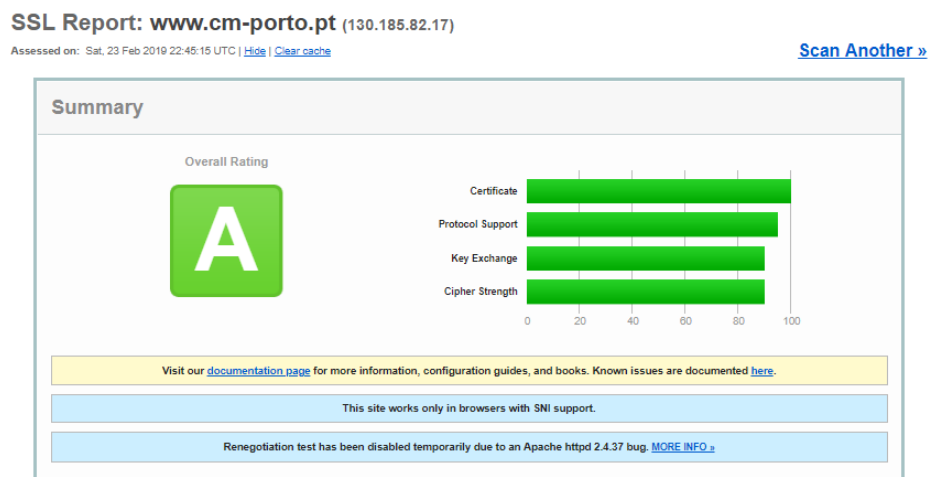


Figura 3: Resultado do SSL Server Test para cm-porto.pt

## 2.1.3 Guimarães

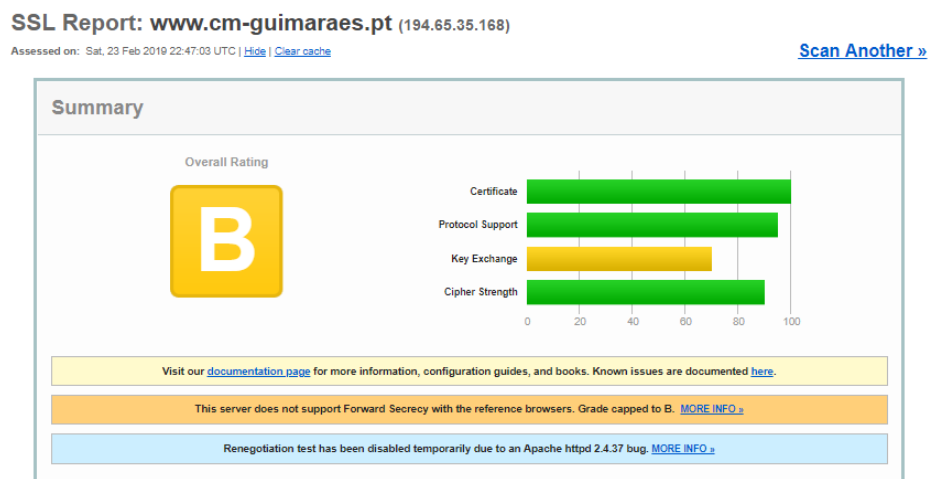


Figura 4: Resultado do SSL Server Test para cm-guimaraes.com

## 2.2 Segurança do site da Câmara Municipal de Braga

Através dos resultados obtidos na questão anterior, foi possível concluir que o site com um pior rating e por isso pior segurança é o site da Câmara Municipal de Braga, *cm-braga.pt*, com um rating de C. Ao analisar os resultados foi possível observar que é utilizado um certificado que expirou há mais de 2 anos e meio, logo não deveria de todo ser utilizado, pois já não se trata de um certificado de confiança.



Certificate #2: RSA 2048 bits (SHA256withRSA)	
 Server Key and Certificate #1	
Subject	*.cm-braga.pt Fingerprint SHA256: cd96444baab4def45961ab3fadbcb852afca53fc5afbe97e681096c5850a03e Pin SHA256: /4+1QqTCGDAKTF3KdUj135Z7p58ec8uRqUP9Jvxxw
Common names	*.cm-braga.pt
Alternative names	*.cm-braga.pt cm-braga.pt
Serial Number	6127c3a3a79b0be0509f05f2cbe8ca2d
Valid from	Fri, 17 Jul 2015 00:00:00 UTC
Valid until	Sat, 16 Jul 2016 23:59:59 UTC (expired 2 years and 7 months ago) <b>EXPIRED</b>
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	COMODO RSA Domain Validation Secure Server CA AIA: http://crl.comodoca.com/COMODORSADomainValidationSecureServerCA.crl
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	No
OCSF Must Staple	No
Revocation information	CRL OCSF CRL: http://crl.comodoca.com/COMODORSADomainValidationSecureServerCA.crl OCSF: http://ocsp.comodoca.com
Revocation status	Unchecked (only trusted certificates can be checked)
DNS CAA	No ( <a href="#">more info</a> )
Trusted	No <b>NOT TRUSTED</b> Mozilla Apple Android Java Windows

Figura 5: Certificado Expirado

Este resultado deve-se em grande parte ao facto do site estar vulnerável a um ataque *POODLE(SSLv3)*, que põe em causa a segurança do site e desce o *rating* logo para C. Para além disso, como é possível observar na seguinte imagem, também utiliza *RC4*, que já não é seguro.



Protocol Details					
	IP Address	Port	Export	Special	Status
	62.28.4.76	443	Yes	No	Not vulnerable
DROWN	(1) For a better understanding of this test, please read <a href="#">this longer explanation</a> (2) Key usage data kindly provided by the <a href="#">Censys</a> network search engine; original DROWN website <a href="#">here</a> (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and incomplete (4) We perform real-time key reuse checks, but stop checking after first confirmed vulnerability (5) The "Special" column indicates vulnerable OpenSSL version; "Export" refers to export cipher suites				
BEAST attack	Not mitigated server-side ( <a href="#">more info</a> )				
POODLE (SSLv3)	Vulnerable	INSECURE ( <a href="#">more info</a> )	SSL 3.0; TLS 1.0; TLS 1.1		
POODLE (TLS)	No ( <a href="#">more info</a> )				
Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported ( <a href="#">more info</a> )				
SSL/TLS compression	No				
RC4	Yes	INSECURE ( <a href="#">more info</a> )			
Heartbeat (extension)	Yes				
Heartbleed (vulnerability)	No ( <a href="#">more info</a> )				
Ticketbleed (vulnerability)	No ( <a href="#">more info</a> )				
OpenSSL CCS vuln. (CVE-2014-0224)	No ( <a href="#">more info</a> )				
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No ( <a href="#">more info</a> )				
ROBOT (vulnerability)	No ( <a href="#">more info</a> )				
Forward Secrecy	With some browsers ( <a href="#">more info</a> )				

Figura 6: Detalhes dos Protocolos Utilizados

## 2.3 POODLE (TLS)

Um ataque *POODLE* direcionado ao TLS procura falhas no modo de cifra CBC. O *padding* utilizado não é devidamente validado nos servidores que desativam SSL 3.0. Como isto não acontece em nenhum dos sites abordados, podemos afirmar que estes se encontram resistentes a este tipo de ataque.

### 3 Protocolo SSH

Empresas localizadas em Madrid que iremos utilizar *ssh-audit* para realizar testes:



<b>77.226.253.213</b> static-213-253-226-77.ipoom.comunitel.net <b>Vodafone Spain</b> Added on 2019-02-23 23:00:01 GMT  Spain, Madrid	SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u5 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAQAC1G+fihaQwxPazXm33iYm/MwU0/58F0VgGyx0lk9vg1ywG QPZF1P++tnLLuoYc6xDzoVrf6+ElDPCEANzBuNtPFF8gzW9E1RJtohEA7fuBMwJ9LYVEYmE804lK ygNMDu+0ShwFqNqgzCQPG0j3YhEEPzZgp9KtYGB1w9itz6CHLreMJQ8sCC57wbffuG6mbg50P0HS SN11...
<b>91.142.211.210</b> recanet.gestimpost.com <b>Axarnet Comunicaciones SL</b> Added on 2019-02-23 23:01:31 GMT  Spain, Madrid	SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze4 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAQDFAYNXuxWcId7MSnCBbo9FyNJqy1fxmHNI1PzKu9aChLvj DUoECkh6mJL0Hg34KcMeFQkpsTK0poMuJcmRIUV+8BvrbWZIXxH8ICFGNQWQfQbccNSxA5CI/CP2 a+XLgLGHa8To6RJB8Q88Kpb8ebA0BKfMzZbKSQNDHUoPDMt1sdUuvQVMabyxNR7q6+f9c8c4vnIc weJ...

Figura 7: Empresas localizadas em Madrid

#### 3.1 Testes ssh de empresas comerciais de Madrid

Através do comando *ssh-audit* foi possível obter os seguinte resultados para estas empresas:

### 3.1.1 Vodafone Spain - 77.226.253.213

```
user@CSI:~/Tools/ssh-audit$ python ssh-audit.py 77.226.253.213
# general
(gen) banner: SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u5
(gen) software: OpenSSH 7.4p1
(gen) compatibility: OpenSSH 7.3+, Dropbear SSH 2016.73+
(gen) compression: enabled (zlib@openssh.com)

# key exchange algorithms
(kex) curve25519-sha256 -- [warn] unknown algorithm
(kex) curve25519-sha256@libssh.org -- [info] available since OpenSSH 6.5, Dropbear SSH 2013.62
(kex) ecdh-sha2-nistp256 -- [fail] using weak elliptic curves
-- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(kex) ecdh-sha2-nistp384 -- [fail] using weak elliptic curves
-- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(kex) ecdh-sha2-nistp521 -- [fail] using weak elliptic curves
-- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(kex) diffie-hellman-group-exchange-sha256 -- [warn] using custom size modulus (possibly weak)
-- [info] available since OpenSSH 4.4
(kex) diffie-hellman-group16-sha512 -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73
(kex) diffie-hellman-group18-sha512 -- [info] available since OpenSSH 7.3
(kex) diffie-hellman-group14-sha256 -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73
(kex) diffie-hellman-group14-sha1 -- [warn] using weak hashing algorithm
-- [info] available since OpenSSH 3.9, Dropbear SSH 0.53

# host-key algorithms
(key) ssh-rsa -- [info] available since OpenSSH 2.5.0, Dropbear SSH 0.28
(key) rsa-sha2-512 -- [info] available since OpenSSH 7.2
(key) rsa-sha2-256 -- [info] available since OpenSSH 7.2
(key) ecdsa-sha2-nistp256 -- [fail] using weak elliptic curves
-- [warn] using weak random number generator could reveal the key
-- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(key) ssh-ed25519 -- [info] available since OpenSSH 6.5

# encryption algorithms (ciphers)
(enc) chacha20-poly1305@openssh.com -- [info] available since OpenSSH 6.5
-- [info] default cipher since OpenSSH 6.9.
(enc) aes128-ctr -- [info] available since OpenSSH 3.7, Dropbear SSH 0.52
(enc) aes192-ctr -- [info] available since OpenSSH 3.7
(enc) aes256-ctr -- [info] available since OpenSSH 3.7, Dropbear SSH 0.52
(enc) aes128-gcm@openssh.com -- [info] available since OpenSSH 6.2
(enc) aes256-gcm@openssh.com -- [info] available since OpenSSH 6.2

# message authentication code algorithms
(mac) umac-64-etm@openssh.com -- [warn] using small 64-bit tag size
-- [info] available since OpenSSH 6.2
(mac) umac-128-etm@openssh.com -- [info] available since OpenSSH 6.2
(mac) hmac-sha2-256-etm@openssh.com -- [info] available since OpenSSH 6.2
(mac) hmac-sha2-512-etm@openssh.com -- [info] available since OpenSSH 6.2
-- [warn] using weak hashing algorithm
(mac) hmac-sha1-etm@openssh.com -- [info] available since OpenSSH 6.2
-- [warn] using encrypt-and-MAC mode
-- [warn] using small 64-bit tag size
-- [info] available since OpenSSH 4.7
(mac) umac-128@openssh.com -- [warn] using encrypt-and-MAC mode
-- [info] available since OpenSSH 6.2
(mac) hmac-sha2-256 -- [warn] using encrypt-and-MAC mode
-- [info] available since OpenSSH 5.9, Dropbear SSH 2013.56
(mac) hmac-sha2-512 -- [warn] using encrypt-and-MAC mode
-- [info] available since OpenSSH 5.9, Dropbear SSH 2013.56
(mac) hmac-sha1 -- [warn] using encrypt-and-MAC mode
-- [warn] using weak hashing algorithm
-- [info] available since OpenSSH 2.1.0, Dropbear SSH 0.28

# algorithm recommendations (for OpenSSH 7.4)
(rec) -ecdh-sha2-nistp521 -- kex algorithm to remove
(rec) -ecdh-sha2-nistp384 -- kex algorithm to remove
(rec) -diffie-hellman-group14-sha1 -- kex algorithm to remove
(rec) -ecdh-sha2-nistp256 -- kex algorithm to remove
(rec) -diffie-hellman-group-exchange-sha256 -- kex algorithm to remove
(rec) -ecdsa-sha2-nistp256 -- key algorithm to remove
(rec) -hmac-sha2-512 -- mac algorithm to remove
(rec) -umac-128@openssh.com -- mac algorithm to remove
(rec) -hmac-sha2-256 -- mac algorithm to remove
(rec) -umac-64@openssh.com -- mac algorithm to remove
(rec) -hmac-sha1 -- mac algorithm to remove
(rec) -hmac-sha1-etm@openssh.com -- mac algorithm to remove
(rec) -umac-64-etm@openssh.com -- mac algorithm to remove

user@CSI:~/Tools/ssh-audit$
```

Figura 8: Resultados obtidos para 77.226.253.213

### 3.1.2 Axarnet Communications - 91.141.211.210

```
user@CSI:~/Tools/ssh-audit$ python ssh-audit.py 91.142.211.210
# general
(gen) banner: SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze4
(gen) software: OpenSSH 5.5p1
(gen) compatibility: OpenSSH 4.7-6.6, Dropbear SSH 0.53+ (some functionality from 0.52)
(gen) compression: enabled (zlib@openssh.com)

# key exchange algorithms
(kex) diffie-hellman-group-exchange-sha256 -- [warn] using custom size modulus (possibly weak)
-- [info] available since OpenSSH 4.4
(kex) diffie-hellman-group-exchange-sha1 -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] using weak hashing algorithm
-- [info] available since OpenSSH 2.3.0
(kex) diffie-hellman-group14-sha1 -- [warn] using weak hashing algorithm
-- [info] available since OpenSSH 3.9, Dropbear SSH 0.53
(kex) diffie-hellman-group1-sha1 -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [fail] disabled (in client) since OpenSSH 7.0, logjam attack
-- [warn] using small 1024-bit modulus
-- [warn] using weak hashing algorithm
-- [info] available since OpenSSH 2.3.0, Dropbear SSH 0.28

# host-key algorithms
(key) ssh-rsa -- [info] available since OpenSSH 2.5.0, Dropbear SSH 0.28
-- [fail] removed (in server) and disabled (in client) since OpenSSH 7.0, weak algorithm
(key) ssh-dss -- [warn] using small 1024-bit modulus
-- [warn] using weak random number generator could reveal the key
-- [info] available since OpenSSH 2.1.0, Dropbear SSH 0.28

# encryption algorithms (ciphers)
(enc) aes128-ctr -- [info] available since OpenSSH 3.7, Dropbear SSH 0.52
-- [info] available since OpenSSH 3.7
(enc) aes192-ctr -- [info] available since OpenSSH 3.7, Dropbear SSH 0.52
(enc) aes256-ctr -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] disabled (in client) since OpenSSH 7.2, legacy algorithm
-- [warn] using weak cipher
-- [info] available since OpenSSH 4.2
(enc) arcfour128 -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] disabled (in client) since OpenSSH 7.2, legacy algorithm
-- [warn] using weak cipher
-- [info] available since OpenSSH 4.2
(enc) aes128-cbc -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] using weak cipher mode
-- [info] available since OpenSSH 2.3.0, Dropbear SSH 0.28
(enc) 3des-cbc -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] using weak cipher
-- [warn] using weak cipher mode
-- [warn] using small 64-bit block size
-- [info] available since OpenSSH 1.2.2, Dropbear SSH 0.28
(enc) blowfish-cbc -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [fail] disabled since Dropbear SSH 0.53
-- [warn] disabled (in client) since OpenSSH 7.2, legacy algorithm
-- [warn] using weak cipher mode
-- [warn] using small 64-bit block size
-- [info] available since OpenSSH 1.2.2, Dropbear SSH 0.28
(enc) cast128-cbc -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] disabled (in client) since OpenSSH 7.2, legacy algorithm
-- [warn] using weak cipher mode
-- [warn] using small 64-bit block size
-- [info] available since OpenSSH 2.1.0
(enc) aes192-cbc -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] using weak cipher mode
-- [info] available since OpenSSH 2.3.0
(enc) aes256-cbc -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] using weak cipher mode
-- [info] available since OpenSSH 2.3.0, Dropbear SSH 0.47
(enc) arcfour -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] disabled (in client) since OpenSSH 7.2, legacy algorithm
-- [warn] using weak cipher
-- [info] available since OpenSSH 2.1.0
(enc) rijndael-cbc@lysator.liu.se -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] disabled (in client) since OpenSSH 7.2, legacy algorithm
-- [warn] using weak cipher mode
-- [info] available since OpenSSH 2.3.0

# message authentication code algorithms
(mac) hmac-md5 -- [fail] removed (in server) since OpenSSH 6.7, unsafe algorithm
-- [warn] disabled (in client) since OpenSSH 7.2, legacy algorithm
-- [warn] using encrypt-and-MAC mode
-- [warn] using weak hashing algorithm
```

Figura 9: Resultados obtidos para 91.141.211.210



## 3.2 Software e Versões utilizadas pelos servidores ssh

O software utilizado pela Vodafone Spain é o *openssh* e utiliza a versão 7.4. O software utilizado pela Axarnet Communications é o *openssh* e utiliza a versão 5.5.p1.

## 3.3 Vulnerabilidades

Como seria de esperar, a versão mais antiga, 5.5.p1 possui mais vulnerabilidades, com 12. Enquanto que a versão 7.4 apenas possui 3 vulnerabilidades.

Openbsd » Openssh » 5.5 : Security Vulnerabilities

Cpe Name:cpe:/a:openbsd:openssh:5.5

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2018-15473</a>	<a href="#">200</a>		+Info	2018-08-17	2018-12-05	5.0	None	Remote	Low	Not required	Partial	None	None
OpenSSH through 7.7 is prone to a user enumeration vulnerability due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed, related to auth2-gss.c, auth2-hostbased.c, and auth2-pubkey.c.														
2	<a href="#">CVE-2017-15906</a>	<a href="#">275</a>			2017-10-25	2018-09-11	5.0	None	Remote	Low	Not required	None	Partial	None
The process_open function in sftp-server.c in OpenSSH before 7.6 does not properly prevent write operations in readonly mode, which allows attackers to create zero-length files.														
3	<a href="#">CVE-2016-10708</a>	<a href="#">476</a>		DoS	2018-01-21	2018-11-07	5.0	None	Remote	Low	Not required	None	None	Partial
sshd in OpenSSH before 7.4 allows remote attackers to cause a denial of service (NULL pointer dereference and daemon crash) via an out-of-sequence NEWKEYS message, as demonstrated by Honggfuzz, related to kex.c and packet.c.														
4	<a href="#">CVE-2016-0778</a>	<a href="#">119</a>		DoS Overflow	2016-01-14	2018-10-09	4.6	None	Remote	High	Single system	Partial	Partial	Partial
The (1) roaming_read and (2) roaming_write functions in roaming_common.c in the client in OpenSSH 5.x, 6.x, and 7.x before 7.1p2, when certain proxy and forward options are enabled, do not properly maintain connection file descriptors, which allows remote servers to cause a denial of service (heap-based buffer overflow) or possibly have unspecified other impact by requesting many forwardings.														
5	<a href="#">CVE-2016-0777</a>	<a href="#">200</a>		+Info	2016-01-14	2018-10-09	4.0	None	Remote	Low	Single system	Partial	None	None
The resend_bytes function in roaming_common.c in the client in OpenSSH 5.x, 6.x, and 7.x before 7.1p2 allows remote servers to obtain sensitive information from process memory by requesting transmission of an entire buffer, as demonstrated by reading a private key.														
6	<a href="#">CVE-2014-1692</a>	<a href="#">119</a>		DoS Overflow Mem. Corr.	2014-01-29	2017-08-28	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
The hash_buffer function in schnorr.c in OpenSSH through 6.4, when Makefile.inc is modified to enable the J-PAKE protocol, does not initialize certain data structures, which might allow remote attackers to cause a denial of service (memory corruption) or have unspecified other impact via vectors that trigger an error condition.														
7	<a href="#">CVE-2012-0814</a>	<a href="#">255</a>		+Info	2012-01-27	2017-08-28	3.5	None	Remote	Medium	Single system	Partial	None	None
The auth_parse_options function in auth-options.c in sshd in OpenSSH before 5.7 provides debug messages containing authorized_keys command options, which allows remote authenticated users to obtain potentially sensitive information by reading these messages, as demonstrated by the shared user account required by Gitolite. NOTE: this can cross privilege boundaries because a user account may intentionally have no shell or filesystem access, and therefore may have no supported way to read an authorized_keys file in its own home directory.														
8	<a href="#">CVE-2011-5000</a>	<a href="#">189</a>		DoS	2012-04-05	2012-07-21	3.5	None	Remote	Medium	Single system	None	None	Partial
The ssh_gssapi_parse_ename function in gss-serv.c in OpenSSH 5.8 and earlier, when gssapi-with-mic authentication is enabled, allows remote authenticated users to cause a denial of service (memory consumption) via a large value in a certain length field. NOTE: there may be limited scenarios in which this issue is relevant.														
9	<a href="#">CVE-2011-4327</a>	<a href="#">200</a>		+Info	2014-02-02	2014-02-21	2.1	None	Local	Low	Not required	Partial	None	None
ssh-keysign.c in ssh-keysign in OpenSSH before 5.8p2 on certain platforms executes ssh-rand-helper with unintended open file descriptors, which allows local users to obtain sensitive key information via the ptrace system call.														
10	<a href="#">CVE-2010-5107</a>			DoS	2013-03-07	2017-09-18	5.0	None	Remote	Low	Not required	None	None	Partial
The default configuration of OpenSSH through 6.1 enforces a fixed time limit between establishing a TCP connection and completing a login, which makes it easier for remote attackers to cause a denial of service (connection-slot exhaustion) by periodically making many new TCP connections.														
11	<a href="#">CVE-2010-4755</a>	<a href="#">399</a>		DoS	2011-03-02	2014-08-08	4.0	None	Remote	Low	Single system	None	None	Partial
The (1) remote_glob function in sftp-glob.c and the (2) process_put function in sftp.c in OpenSSH 5.8 and earlier, as used in FreeBSD 7.3 and 8.1, NetBSD 5.0.2, OpenBSD 4.7, and other products, allow remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, as demonstrated by glob expressions in SSH_FXP_STAT requests to an sftp daemon, a different vulnerability than CVE-2010-2632.														
12	<a href="#">CVE-2010-4478</a>	<a href="#">287</a>		Bypass	2010-12-06	2017-09-18	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
OpenSSH 5.6 and earlier, when J-PAKE is enabled, does not properly validate the public parameters in the J-PAKE protocol, which allows remote attackers to bypass the need for knowledge of the shared secret, and successfully authenticate, by sending crafted values in each round of the protocol, a related issue to CVE-2010-4252.														

Total number of vulnerabilities : 12 Page : 1 (This Page)

Figura 10: Vulnerabilidades da versão 5.5

## Openbsd » Openssh » 7.4 P1 : Security Vulnerabilities

Cpe Name: *cpe:/a:openbsd:openssh:7.4:p1*

CVSS Scores Greater Than: [0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)

Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2018-15919</a>	<a href="#">200</a>		+Info	2018-08-28	2018-12-22	5.0	None	Remote	Low	Not required	Partial	None	None
Remotely observable behaviour in auth-gss2.c in OpenSSH through 7.8 could be used by remote attackers to detect existence of users on a target system when GSS2 is in use. NOTE: the discoverer states 'We understand that the OpenSSH developers do not want to treat such a username enumeration (or "oracle") as a vulnerability.'														
2	<a href="#">CVE-2018-15473</a>	<a href="#">200</a>		+Info	2018-08-17	2018-12-05	5.0	None	Remote	Low	Not required	Partial	None	None
OpenSSH through 7.7 is prone to a user enumeration vulnerability due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed, related to auth2-gss.c, auth2-hostbased.c, and auth2-pubkey.c.														
3	<a href="#">CVE-2017-15906</a>	<a href="#">275</a>			2017-10-25	2018-09-11	5.0	None	Remote	Low	Not required	None	Partial	None

The process\_open function in sftp-server.c in OpenSSH before 7.6 does not properly prevent write operations in readonly mode, which allows attackers to create zero-length files.

Total number of vulnerabilities : **3** Page : [1](#) (This Page)

Figura 11: Vulnerabilidades da versão 7.4

### 3.3.1 Vulnerabilidade mais grave

Existem duas vulnerabilidades que possuem um *CVSS Score* de 7.5.

#### Vulnerability Details : [CVE-2014-1692](#)

The hash\_buffer function in schnorr.c in OpenSSH through 6.4, when Makefile.inc is modified to enable the J-PAKE protocol, does not initialize certain data structures, which might allow remote attackers to cause a denial of service (memory corruption) or have unspecified other impact via vectors that trigger an error condition.

Publish Date : 2014-01-29 Last Update Date : 2017-08-28

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [Scroll To](#) [Comments](#) [External Links](#)

[Search Twitter](#) [Search YouTube](#) [Search Google](#)

#### - CVSS Scores & Vulnerability Types

CVSS Score	7.5
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit. )
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Denial Of Service Overflow Memory corruption
CWE ID	<a href="#">119</a>

Figura 12: CVE-2014-1692

#### Vulnerability Details : [CVE-2010-4478](#)

OpenSSH 5.6 and earlier, when J-PAKE is enabled, does not properly validate the public parameters in the J-PAKE protocol, which allows remote attackers to bypass the need for knowledge of the shared secret, and successfully authenticate, by sending crafted values in each round of the protocol, a related issue to CVE-2010-4252.

Publish Date : 2010-12-06 Last Update Date : 2017-09-18

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [Scroll To](#) [Comments](#) [External Links](#)  
[Search Twitter](#) [Search YouTube](#) [Search Google](#)

#### – CVSS Scores & Vulnerability Types

CVSS Score	<b>7.5</b>
Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit. )
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Bypass a restriction or similar
CWE ID	<a href="#">287</a>

Figura 13: CVE-2010-4478

### 3.3.2 Gravidade das Vulnerabilidades

Ambas as vulnerabilidades são bastante graves, daí a sua classificação de 7.5, no entanto, a que permite ultrapassar o método de autenticação de segredo partilhado é mais preocupante, pois permite ao atacante aceder por completo a este serviço sem os devidos requerimentos. Apesar da outra vulnerabilidade também ser grave, apenas afeta o funcionamento do serviço, impedindo o funcionamento deste não comprometendo a sua integridade.