

Mestrado em Engenharia Informática  
Universidade do Minho

**Engenharia de Segurança**

**Aula 03 TP - 03/03/2020**

João Miranda - PG41845  
Sandro Cruz - PG41906

6 de Março de 2020

# Conteúdo

<b>1</b>	<b>Assinaturas cegas (Blind signatures) baseadas no Elliptic Curve Discrete Logarithm Problem (ECDLP)</b>	<b>2</b>
1.1	Experiência 1.1 . . . . .	2
1.2	Experiência P1.2 . . . . .	3
1.3	Experiência 1.3 . . . . .	4
1.3.1	Script init-app.py . . . . .	4
1.3.2	Script ofusca-app.py . . . . .	4
1.3.3	Script blindSignature-app.py . . . . .	5
1.3.4	Script desofusca-app.py . . . . .	5
1.3.5	Script verify-app.py . . . . .	5
<b>2</b>	<b>Protocolo SSL/TLS</b>	<b>6</b>
2.1	Experiência 2.1 . . . . .	6
2.2	Pergunta P2.1 . . . . .	7
<b>3</b>	<b>Protocolo SSH</b>	<b>10</b>
3.1	Experiência 3.1 . . . . .	10
3.2	Pergunta P3.1 . . . . .	10

# Capítulo 1

## Assinaturas cegas (Blind signatures) baseadas no Elliptic Curve Discrete Logarithm Problem (ECDLP)

### 1.1 Experiência 1.1

```
openssl ecparam -name prime256v1 -genkey -noout -out key.pem
```

```
user@CSI:~/Aulas/Aula3/BlindSignature$ openssl ecparam -name prime256v1 -genkey  
-noout -out key.pem
```

Figura 1.1: Assinaturas Cegas - Output do comando acima

```
openssl req -key key.pem -new -x509 -days 365 -out key.crt
```

```
user@CSI:~/Aulas/Aula3/BlindSignature$ openssl req -key key.pem -new -x509 -days  
365 -out key.crt  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Country Name (2 letter code) [AU]:PT  
State or Province Name (full name) [Some-State]:Braga  
Locality Name (eg, city) []:Braga  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UM  
Organizational Unit Name (eg, section) []:MEIE  
Common Name (e.g. server FQDN or YOUR name) []:MEIE  
Email Address []:.
```

Figura 1.2: Assinaturas Cegas - Output do comando acima

## 1.2 Experiência P1.2

`python initSigner-app.py`

```
user@CSI:~/Aulas/Aula3/BlindSignature$ python initSigner-app.py
Output
Init components: 259363f20c046a22f09ee2ba8fe5e89db2ab5abc8fbb879512bf44f3bcc690aa.6d4fccbb84b7e98db1b8a9f9c0c7c8ddd6b9c9bab2e0e6744a6c
d90a6160b075
pRDashComponents: 259363f20c046a22f09ee2ba8fe5e89db2ab5abc8fbb879512bf44f3bcc690aa.e3fcc8b63e062240a6251fddbb2a1fa66b22ac57bf9974154e3
53f3fca58b0e9
```

Figura 1.3: Assinaturas Cegas - Inicialização da Assinatura

`python generateBlindData-app.py`

```
user@CSI:~/Aulas/Aula3/BlindSignature$ python generateBlindData-app.py
Input
Data: Uma mensagem
pRDash components: 259363f20c046a22f09ee2ba8fe5e89db2ab5abc8fbb879512bf44f3bcc690aa.e3fcc8b63e062240a6251fddbb2a1fa66b22ac57bf9974154e
353f3fca58b0e9
Output
Blind message: f309abbc71a0bee2dd7eae32900d1ea970a89a2ecf208160b0770bf653cde5b9
Blind components: 25958337b40f86ad41f92add32a9c3982652119c2ealbeef4564cb5bfe5dc95f.cc6e2799d255281e75d05626fc21380c770ef7d107042a44f6d
aa50967906368
pRComponents: cc6e2799d255281e75d05626fc21380c770ef7d107042a44f6daa50967906368.863071079cc49d7bc1d77eb658a25e539ffe0a62ed1fc6d95fb6448
d2cef67d1
```

Figura 1.4: Assinaturas Cegas - Ofuscação da Mensagem

`python generateBlindSignature-app.py key.pem`

```
user@CSI:~/Aulas/Aula3/BlindSignature$ python generateBlindSignature-app.py key.pem
Input
Passphrase:
Blind message: f309abbc71a0bee2dd7eae32900d1ea970a89a2ecf208160b0770bf653cde5b9
Init components: 259363f20c046a22f09ee2ba8fe5e89db2ab5abc8fbb879512bf44f3bcc690aa.6d4fccbb84b7e98db1b8a9f9c0c7c8ddd6b9c9bab2e0e6744a6c
d90a6160b075
Output
Blind signature: 94ca30e031f8e2ebd5116f24ba8541838485465109c3845d4b583cbc0ad42ce59e2f05363be74c53076184f0f70f153d4a8399c1571d4c4b154d4
ed39d5020d2
```

Figura 1.5: Assinaturas Cegas - Assinatura da Mensagem Ofuscado por uma Entidade

`python unblindSignature-app.py`

```
user@CSI:~/Aulas/Aula3/BlindSignature$ python unblindSignature-app.py
Input
Blind signature: 94ca30e031f8e2ebd5116f24ba8541838485465109c3845d4b583cbc0ad42ce59e2f05363be74c53076184f0f70f153d4a8399c1571d4c4b154d4
ed39d5020d2
Blind components: 25958337b40f86ad41f92add32a9c3982652119c2ealbeef4564cb5bfe5dc95f.cc6e2799d255281e75d05626fc21380c770ef7d107042a44f6d
aa50967906368
pRDash components: 259363f20c046a22f09ee2ba8fe5e89db2ab5abc8fbb879512bf44f3bcc690aa.e3fcc8b63e062240a6251fddbb2a1fa66b22ac57bf9974154e
353f3fca58b0e9
Output
Signature: 2c50a8cd5a6441da75b9a32319fc5ca4df7cee8c3fd5d6ece928ab9cf6d9b501
```

Figura 1.6: Assinaturas Cegas - Desofuscação da Mensagem

`python verifySignature-app.py key.crt`

```

user@CSI:~/Aulas/Aula3/BlindSignature$ python verifySignature-app.py key.crt
Input
Original data: Uma mensagem
Signature: 2c50a8cd5a6441da75b9a32319fc5ca4df7cee8c3fd5d6ece928ab9cf6d9b501
Blind components: 25958337b40f86ad41f92add32a9c3982652119c2ea1beef4564cb5bfe5dc95f.cc6e2799d255281e75d05626fc21380c770ef7d107042a44f6d
aa50967906368
pR components: cc6e2799d255281e75d05626fc21380c770ef7d107042a44f6daa50967906368.863071079cc49d7bc1d77eb658a25e539ffe0a62ed1fc6d95fb644
8d2cef67d1
Output
Valid signature

```

Figura 1.7: Assinaturas Cegas - Verificação da Assinatura

## 1.3 Experiência 1.3

Todos os *scripts* desenvolvidos para a experiência, vão estar disponíveis no repositório do *git* em anexo.

### 1.3.1 Script init-app.py

```

user@CSI:~/Aulas/Aula3$ python init-app.py -init
user@CSI:~/Aulas/Aula3$

```

Figura 1.8: Execução do Script init-app.py

```

user@CSI:~/Aulas/Aula3$ cat initFile.txt
1b05b5ed9909ce985f7cbd73ab2dcd6fad439ff34cc1a625fb5c625bd49f6a4.698e5764e14bb55dbca02ccf6134fac89299ccb696ec51
cb6716aa8c7f4ddce4
1b05b5ed9909ce985f7cbd73ab2dcd6fad439ff34cc1a625fb5c625bd49f6a4.f2fea2ef965430023eff9f39d761e4e738d53a09fc036e
0a820b217218dcbe1

```

Figura 1.9: Conteúdo do Ficheiro Gerado pelo Script Executado Acima

### 1.3.2 Script ofusca-app.py

```

user@CSI:~/Aulas/Aula3$ python ofusca-app.py -msg Mensagem -RDash 1b05b5ed9909ce985f7cbd73ab2dcd6fad439f
f34cc1a625fb5c625bd49f6a4.f2fea2ef965430023eff9f39d761e4e738d53a09fc036e0a820b217218dcbe1
Output
Blind message: 93715f58c1c460d5cdf447fd98bf6598e1d18c9fdd0b58ec0a94b28c76f8276c
Blind components: d34d7973fb2c0ef362105556df4362bce668400fb6562143996de15df14ec8bc.e9e3ce43351de0fe4fbe5
31baaf473ffeff69e1d4f8bf5d04ff6c8f212f1fbb
pRComponents: e9e3ce43351de0fe4fbe531baaf473ffeff69e1d4f8bf5d04ff6c8f212f1fbb.7897ad276ffd01627ff3e899e
4e3afce50e80703ae63b4fc9e15cfcac4aa317

```

Figura 1.10: Output do Script ofusca-app.py

### 1.3.3 Script blindSignature-app.py

```
user@CSI:~/Aulas/Aula3$ python blindSignature-app.py -key BlindSignature/key.pem -bmsg 93715f58c1c460d5c
df447fd98bf6598e1d18c9fdd0b58ec0a94b28c76f8276c
Input
Passphrase:
Init components: 1b05b5ed9909ce985f7cbd73ab2dcd6fad439ff34cc1a625fb5c625bd49f6a4.698e5764e14bb55dbca02cc
f6134fac89299ccb696ec51cb6716aa8c7f4ddce4
Output
Blind signature: 5a440a4eaae9908417e18ca20e54f500d20cbc7db1a0998ca93ce673c291dbe5ec5ef49209979f6759b5729
4d27eaea06ff2ff8c322371785dbf962e18299184
```

Figura 1.11: Output do Script blindSignature-app.py

### 1.3.4 Script desofusca-app.py

```
user@CSI:~/Aulas/Aula3$ python desofusca-app.py -s 70c137315f1ca34303bcedc9086b29be265a1ec6378440977eea5
7c3b3d0b99d3c7c2c17ab1ad3901ecef86bd9e92498f52cd900f0ce2e83e8d2ebd1f858bac -RDash 6f293895af0d1930c860f
b9fb307165595c7f9ce61546e6c1b04a6f40a7354f1f.af12b9aeb1e86ac77074dabda5866d8007d0559044324e1bd2fed3fbb6c1
017c
Input
Blind components: 38c1ffecac84081d6597dbf74541a45ccc8fab78cfe8c6500711ecae766c7fd3.17e9fc9e36c79ee73da72
9df8db20b1178ba26e2b70b03738de3a5457fbc814
Output
Signature: 2256b42a73c8d42d41e85bdd42e846c6cd7dfed1f4e547cc825c8ca72d60290d
```

Figura 1.12: Output do Script desofusca-app.py

### 1.3.5 Script verify-app.py

```
user@CSI:~/Aulas/Aula3$ python verify-app.py -cert BlindSignature/key.crt -msg Mensagem -sDash a
b8f8a016732fd208d69f1d9486e30342860260fab324c57835deb0952506791 -f ofuscaFile.txt
Output
Valid signature _
```

Figura 1.13: Output do Script verify-app.py

## Capítulo 2

# Protocolo SSL/TLS

### 2.1 Experiência 2.1

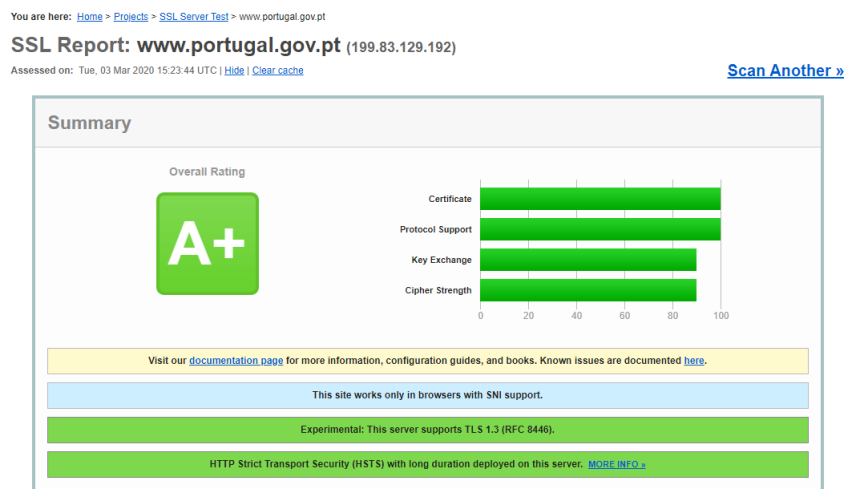


Figura 2.1: SSL Server Test do site do Governo Português.

O resultado do scan completo é possível ser encontrado na pasta **SSL Server Test**. É possível a verificação da classificação que é extremamente positiva (A+) sendo que a segurança é indicada para este site. O servidor suporta o TLS 1.3. É informado também que o HSTS tem uma longa duração quando despoletado no servidor. Este HSTS (HTTP Strict Transport Security) é um mecanismo de política de segurança da web que tem como objetivo ajudar a proteger os websites contra ataques de downgrade de protocolo e do roubo dos cookies. Permite assim que os servidores da web declarem que os utilizadores devam interagir com o website usando apenas conexões HTTPS que fornecem

TLS/SSL.

## 2.2 Pergunta P2.1

Como o nosso grupo é o número 10, a proposta atribuída foi de escolher dois sites de Bancos a operar fora da Europa, isto é, sites com domínios não europeus. Os Bancos que optamos para o SSL Server Test foram: um **Banco do Brasil** denominado **Banco do Nordeste** e um **Banco dos EUA** denominado **JPMorgan Chase**.

i. Anexe os resultados do SSL Server test à sua resposta.

De lembrar que o resultado do scan completo é possível ser encontrado na pasta SSL Server Test.

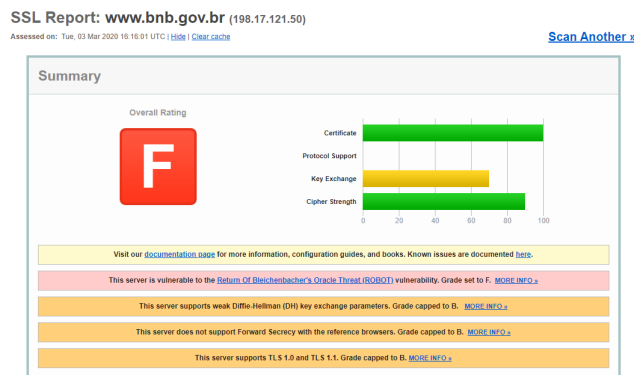


Figura 2.2: SSL Server Test do site do Banco do Nordeste.

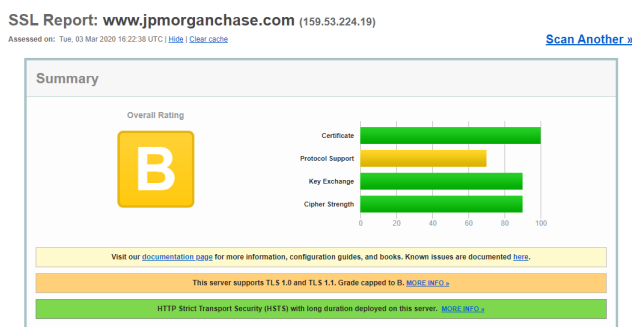


Figura 2.3: SSL Server Test do site do Banco JPMorgan Chase.



ii. Analise o resultado do SSL Server test relativo ao site escolhido com pior rating. Que comentários pode fazer sobre a sua segurança. Porquê?

O Banco que teve piores resultados no SSL Server test foi **Banco do Nordeste** do Brasil com a classificação F. Foram apresentados 4 problemas relativamente à segurança:

- **"This server is vulnerable to the Return Of Bleichenbacher's Oracle Threat (ROBOT) vulnerability. Grade set to F."**

Este problema afirma que o host remoto é afetado por uma vulnerabilidade de divulgação de informações. O serviço SSL/TLS suporta trocas de chaves RSA e dispõe incorretamente se a troca dessas chaves enviada por um cliente foi formatada corretamente ou não. Essas informações podem permitir que um invasor decifre as sessões SSL/TLS anteriores ou represente a identidade do servidor.

- **"This server supports weak Diffie-Hellman (DH) key exchange parameters. Grade capped to B."**

Os parâmetros relativos ao acordo de chaves Diffie-Hellman são fracos, dando possibilidade a que ataques Logjam ocorram, comprometendo a segurança do site. Existe uma possibilidade de ser efetuado um ataque Man-in-the-middle.

- **"This server does not support Forward Secrecy with the reference browsers."**

O **Forward Secrecy (FS)** é uma propriedade de protocolos de comunicação seguros nos quais comprometimentos de chaves de longo prazo não comprometem chaves de sessões anteriores. O sigilo de encaminhamento protege as sessões anteriores contra futuros comprometimentos da chave privada. A troca de chaves RSA não fornece sigilo direto, logo é necessário fornecer suporte e dar preferência aos pacotes ECDHE para permitir sigilo direto nos navegadores de web modernos.

- **"This server supports TLS 1.0 and TLS 1.1."**

Este problema também foi detetado no Banco dos EUA. Isto deve-se aos protocolos TLS 1.0 e 1.1 que irão ser removidos dos browsers no presente ano. Como ainda não existem correções ou patches que possam corrigir adequadamente o SSL ou o TLS obsoleto, é extremamente importante que as organizações atualizem para uma alternativa mais segura, isto é, adotar o protocolo TLS 1.2+.

**iii. É natural que tenha reparado na seguinte informação: "Ticketbleed (vulnerability)" na secção de detalhe do protocolo. O que significa, para efeitos práticos?**

A Ticketbleed é uma vulnerabilidade divulgada recentemente em alguns balanceadores de carga F5. Esses problemas permitem que os invasores recuperem até 31 bytes de memória de processo, isto significa que, que ao sofrermos um ataque por meio desta vulnerabilidade, poderia incluir dados confidenciais (por exemplo, chaves privadas). É de natureza semelhante ao Heartbleed (uma vulnerabilidade no OpenSSL a partir de 2014), mas menos grave porque muito menos dados podem ser extraídos. Por isso deve-se adoptar mecanismos de defesa que permitam evitar ataques provocados por estas vulnerabilidades. Portanto, (R) Indica um navegador de referência ou cliente, com o qual esperamos uma segurança efetiva melhor. São sempre utilizamos padrões, mas algumas plataformas não usam os seus melhores protocolos e recursos (por exemplo, Java 6 e 7, IE mais antigo). A confiança do certificado não é verificada na simulação de handshake, é só executado o handshake TLS.

# Capítulo 3

## Protocolo SSH

### 3.1 Experiência 3.1

```
user@CSI:~/Tools/ssh-audit/Tools/ssh-audit$ python ssh-audit.py algo.paranoidjasmine.com
# general
(gen) banner: SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u7
(gen) software: OpenSSH 7.4p1
(gen) compatibility: OpenSSH 7.3+, Dropbear SSH 2016.73+
(gen) compression: enabled (zlib@openssh.com)

# key exchange algorithms
(key) curve25519-sha256 -- [warn] unknown algorithm
(key) curve25519-sha256@libssh.org -- [info] available since OpenSSH 6.5, Dropbear SSH 2013.62
(key) diffie-hellman-group16-sha512 -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73
(key) diffie-hellman-group18-sha512 -- [info] available since OpenSSH 7.3
(key) diffie-hellman-group14-sha256 -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73

# 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

# 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-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

# algorithm recommendations (for OpenSSH 7.4)
(rec) +ssh-ed25519 -- key algorithm to append
```

Figura 3.1: Resultado do teste ao servidor algo.paranoidjasmine.com com o ssh-audit.

### 3.2 Pergunta P3.1

Com o objetivo de investigar mais acerca do ssh-audit para efetuar testes a websites, o nosso grupo ficou com os servidores ssh (na porta 22) de empresas comerciais em San Francisco.

Primeiramente, foi feita uma pesquisa no website <https://www.shodan.io/> de

servidores na porta 22 em San Francisco. Na figura abaixo são demonstrados os resultados obtidos.

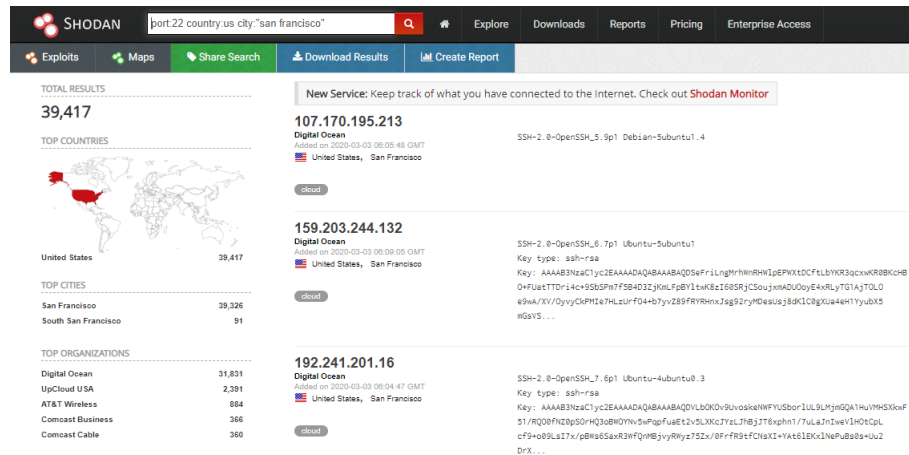


Figura 3.2: Resultado da pesquisa em shodan.

De seguida, foram escolhidas duas empresas comerciais sendo estas a **Digital Ocean** e a **UpCloud USA**.


 <b>159.203.244.132</b>	<a href="#">View Raw Data</a>
<a href="#">cloud</a>	
City	San Francisco
Country	United States
Organization	Digital Ocean
ISP	Digital Ocean
Last Update	2020-03-03T06:09:05.505169
ASN	AS14061

Figura 3.3: Empresa Digital Ocean.

🌐 209.50.48.149 209-50-48-149.us-chi1.upcloud.host [View Raw Data](#)

self-signed

City	San Francisco
Country	United States
Organization	UpCloud USA
ISP	UpCloud USA
Last Update	2020-03-03T06:01:44.208317
Hostnames	209-50-48-149.us-chi1.upcloud.host
ASN	AS25697

Figura 3.4: Empresa UpCloud USA.

Após a pesquisa de duas empresas comerciais, foi utilizado o ssh-audit para efetuar os testes. Os resultados vão ser dispostos de seguida.

### Digital Ocean - 159.203.244.132

```
user@CSI:~/Tools/ssh-audit/Tools/ssh-audit$ python ssh-audit.py 159.203.244.132
# general
(gen) banner: SSH-2.0-OpenSSH 6.7p1 Ubuntu-Subuntul
(gen) software: OpenSSH 6.7p1
(gen) compatibility: OpenSSH 6.5-6.9, Dropbear SSH 2013.62+
(gen) compression: enabled (zlib@openssh.com)

# key exchange algorithms
(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-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) ssh-dss -- [fail] removed (in server) and disabled (in client) since OpenSSH 7.0, weak algorithm
-- [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
(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) 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
(enc) chacha20-poly1305@openssh.com -- [info] available since OpenSSH 6.5
-- [info] default cipher since OpenSSH 6.9.
```

Figura 3.5: ssh-audit - Empresa Digital Ocean, Parte 1.

```
# 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
(mac) hmac-sha1-etm@openssh.com -- [warn] using weak hashing algorithm
-- [info] available since OpenSSH 6.2
(mac) umac-64@openssh.com -- [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 6.7)
(rec) -ecdh-sha2-nistp521 -- kex algorithm to remove
(rec) -ecdh-sha2-nistp384 -- kex algorithm to remove
(rec) -ecdh-sha2-nistp256 -- kex algorithm to remove
(rec) -diffie-hellman-group14-sha1 -- kex algorithm to remove
(rec) -ecdsa-sha2-nistp256 -- key algorithm to remove
(rec) -ssh-dss -- 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
```

Figura 3.6: ssh-audit - Empresa Digital Ocean, Parte 2.

- Software e versão utilizada pelos servidores ssh: OpenSSH 6.7p1
- Resultados da pesquisa

Openbsd » Openssh » 6.7.P1: Security Vulnerabilities

Cpe Name: cpe:/a:openbsd:openssh:6.7.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>	200		+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-2017-13306</a>	269			2017-10-25	2019-10-02	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 read-only mode, which allows attackers to create zero-length files.														
3	<a href="#">CVE-2016-10708</a>	476		DoS	2016-01-21	2019-06-26	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>	119		DoS Overflow	2016-01-14	2018-10-09	4.0	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>	200		+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.														

Total number of vulnerabilities: 5 Page: 1 (This Page)

Figura 3.7: Pesquisa em CVE details do OpenSSH 6.7.

## UpCloud USA - 209.50.48.149

```
user@CSI:~/Tools/ssh-audit/Tools/ssh-audit$ python ssh-audit.py 209.50.48.149
# general
(gen) banner: SSH-2.0-OpenSSH 7.6p1 Ubuntu-4ubuntu0.3
(gen) software: OpenSSH 7.6p1
(gen) compatibility: OpenSSH 7.3+, Dropbear SSH 2016.73+
(gen) compression: enabled (zlib@openssh.com)

# key exchange algorithms
(key) curve25519-sha256 -- [warn] unknown algorithm
(key) curve25519-sha256@libssh.org -- [info] available since OpenSSH 6.5, Dropbear SSH 2013.62
(key) ecdh-sha2-nistp256 -- [fail] using weak elliptic curves
(key) ecdh-sha2-nistp384 -- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(key) ecdh-sha2-nistp521 -- [fail] using weak elliptic curves
(key) diffie-hellman-group-exchange-sha256 -- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(key) diffie-hellman-group16-sha512 -- [warn] using custom size modulus (possibly weak)
(key) diffie-hellman-group18-sha512 -- [info] available since OpenSSH 4.4
(key) diffie-hellman-group14-sha256 -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73
(key) diffie-hellman-group14-sha1 -- [info] available since OpenSSH 7.3
(key) diffie-hellman-group14-sha1 -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73
(key) diffie-hellman-group14-sha1 -- [warn] using weak hashing algorithm
(key) diffie-hellman-group14-sha1 -- [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
(key) ecdsa-sha2-nistp256 -- [warn] using weak random number generator could reveal the key
(key) ssh-ed25519 -- [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
(enc) chacha20-poly1305@openssh.com -- [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
```

Figura 3.8: ssh-audit - Empresa UpCloud USA, Parte 1.

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

# algorithm recommendations (for OpenSSH 7.6)
(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
```

Figura 3.9: ssh-audit - Empresa UpCloud USA, Parte 2.

- Software e versão utilizada pelos servidores ssh: OpenSSH 7.6p1
- Resultados da pesquisa:

Openbsd » Openssh » 7.6 P1: Security Vulnerabilities

Cpe Name: cpe:/a:openbsd:openssh:7.6: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

[Cpev 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	CVE-2018-15919	200		+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.'

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

Figura 3.10: Pesquisa em CVE details do OpenSSH 7.6.

### Qual dessas versões de software tem mais vulnerabilidades?

A versão que possui mais vulnerabilidades é a mais antiga, ou seja, o OpenSSH 6.7.

### E qual tem a vulnerabilidade mais grave (de acordo com o CVSS score identificado no CVE details)?

Ambos têm a vulnerabilidade mais grave com o mesmo valor de 5.0. Logo o OpenSSH 6.7 não tem vulnerabilidade maior que em OpenSSH 7.6 e vice-versa.

### Para efeitos práticos, a vulnerabilidade indicada no ponto anterior é grave? Porquê?

A vulnerabilidade comum às duas, que é de valor 5.0 consiste no comportamento observável remotamente no auth-gss2.c no OpenSSH até ao 7.8 (logo é comum aos dois). Este pode ser usado por atacantes remotos para detetar a existência de utilizadores num sistema de destino quando o GSS2 está em uso.