Aula 3 - 19 de fevereiro de 2018

Grupo 8

1. Assinaturas cegas baseadas no ECDLP

Pergunta P1.1

Relativamente ao assinante, tínhamos as seguintes alterações a fazer:

- Ao executar o comando “python initSigner-app.py -init”, são inicializadas as componentes initComponents e pRDashComponents e guardadas no ficheiro “outputSigner.txt”. Se for apenas executado o ficheiro “initSigner-app.py”, assume-se que as componentes já foram inicializadas, e é apenas imprimido o valor de pRDashComponents, sendo este lido do ficheiro “outputSigner.txt”.

- Ao executar o comando “python blindSignature-app.py -key <chave privada> -bmsg <Blind message>”, é devolvida a BlindSignature. A componente initComponents é lida a partir do ficheiro “outputSigner.txt”.

No que toca ao requerente, as alterações foram as seguintes:

- Ao executar o comando “generateBlindData-app.py -msg msgtosign -RDash pRDashComponents”, é devolvida a BlindMessage e as restantes componentes são guardadas no ficheiro “outputRequester.txt”.

- Ao executar “unblindSignature-app.py -s <Blind Signature> -RDash <pRDashComponents>”, é devolvida a assinatura.

Quanto ao verificador, ao ser executado o comando “verifySignature-app.py -cert <certificado do assinante> -msg msgoriginal -sDash sig -f <ficheiro do requerente>”, deverá ser indicado se a assinatura sDash sobre a mensagem original msg é válida ou não.

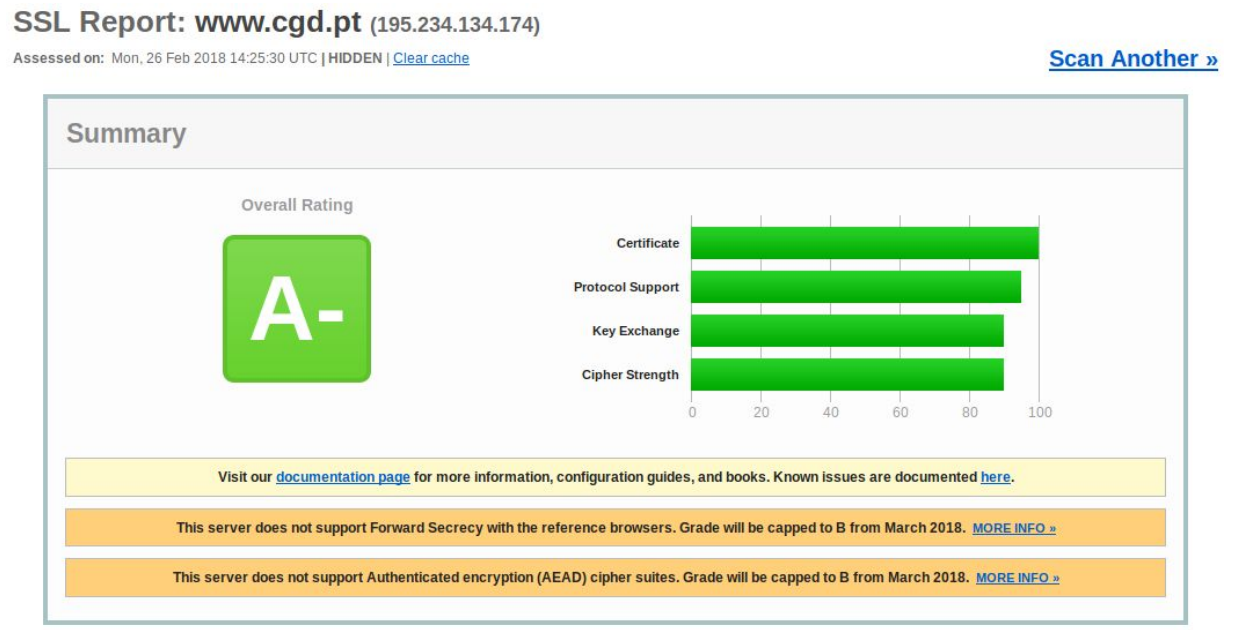
Apresentamos um exemplo de utilização destes comandos:

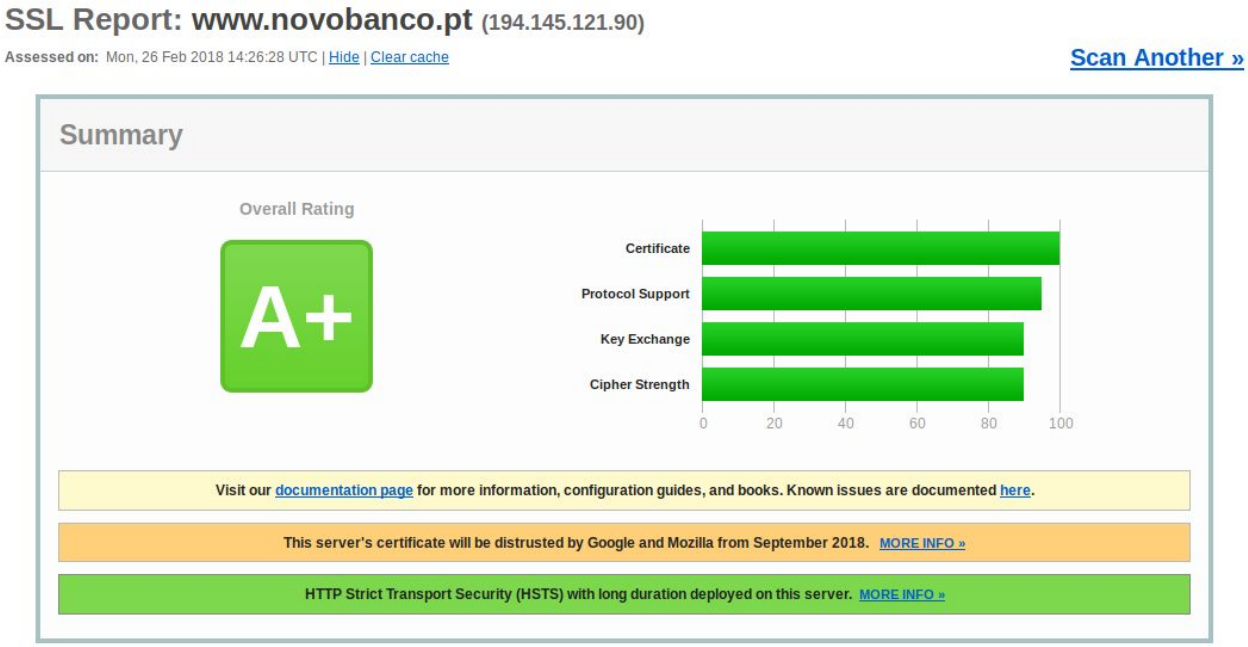
```
user@CSI:~/Aulas/Aula3/BlindSignature$ python initSigner-app.py -init
user@CSI:~/Aulas/Aula3/BlindSignature$ python initSigner-app.py
Output
pRDashComponents: 8ae42ed6e9c33a7b54ace0adb41c530d7753493fa287a8f5596
cc848e44c4bcd.699fe29730d670944028f9521d853be21f9161fabbd86dce89dfaf3
c8dcf61b9
user@CSI:~/Aulas/Aula3/BlindSignature$ python generateBlindData-app.p
y -msg 'Mensagem Secreta' -RDash 8ae42ed6e9c33a7b54ace0adb41c530d7753
493fa287a8f5596cc848e44c4bcd.699fe29730d670944028f9521d853be21f9161fa
bbd86dce89dfaf3c8dcf61b9
Output
Blind message: f5fe1d38b730f990055ab7042da9e2379679872894b2a8998aa563
0abcd2118d
user@CSI:~/Aulas/Aula3/BlindSignature$ python generateBlindSignature-
app.py -key key.pem -bmsg f5fe1d38b730f990055ab7042da9e2379679872894b
2a8998aa5630abcd2118d
Input
Passphrase: 53cr37
Output
Blind signature: 67f21fdfa1345823817a552d19062a71c8e433a88d46c70851d4
5b66c5ec753691b1e105e404fa69876296bc71f860fb75c5c8973bad198246ec04b4f
cf2a49d
user@CSI:~/Aulas/Aula3/BlindSignature$ python unblindSignature-app.py
 -s 67f21fdfa1345823817a552d19062a71c8e433a88d46c70851d45b66c5ec75369
1b1e105e404fa69876296bc71f860fb75c5c8973bad198246ec04b4fcf2a49d -RDas
h 8ae42ed6e9c33a7b54ace0adb41c530d7753493fa287a8f5596cc848e44c4bcd.69
9fe29730d670944028f9521d853be21f9161fabbd86dce89dfaf3c8dcf61b9
Output
Signature: 389b1fcd19d89de920c1f7ae294198bffb1a07d899470467195e8ba401
890fad
user@CSI:~/Aulas/Aula3/BlindSignature$ python verifySigature-app.py -
cert key.crt -msg 'Mensagem Secreta' -sDash 389b1fcd19d89de920c1f7ae2
94198bffb1a07d899470467195e8ba401890fad -f outputRequester.txt
Output
Valid signature
```

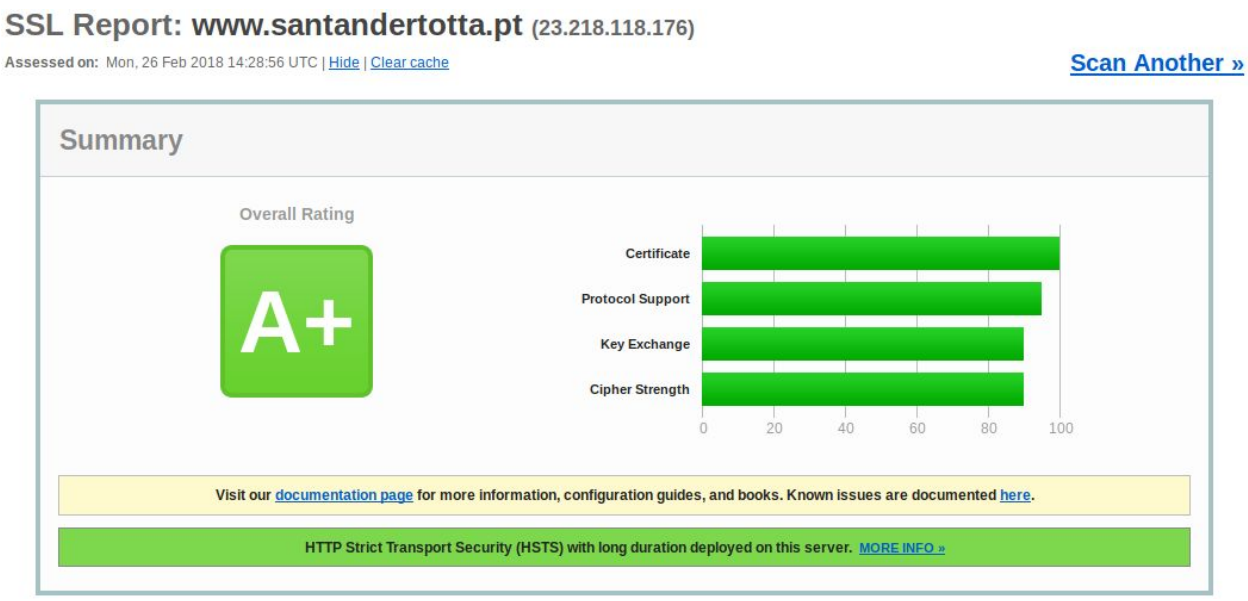
2. Protocolo SSL/TLS

Pergunta P2.1

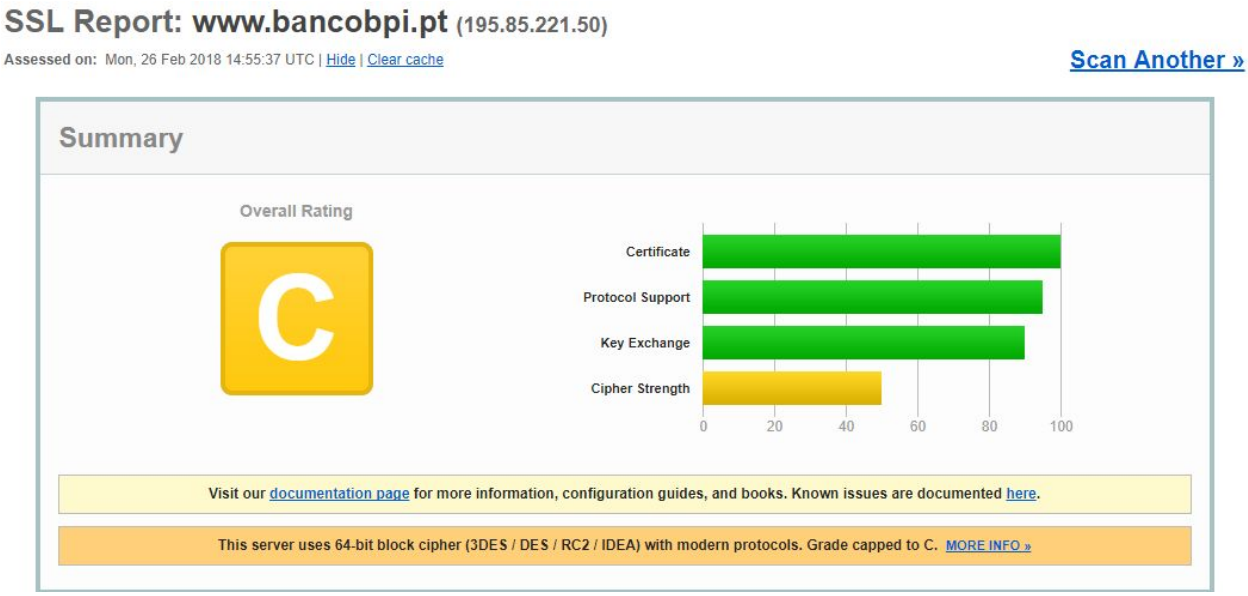
a) Os bancos que escolhemos para analisar foram: CGD, Novo Banco, Santander Totta e BPI. Os três primeiros apresentaram resultados semelhantes entre si:







Já o último teve o site com a pior classificação:



b) A maior vulnerabilidade do site é o facto de usar cifras de blocos com 64-bit (3DES), sendo estas vulneráveis ao *birthday attack* (CVE-2016-2183, CVE-2016-6329).

c) O “Downgrade attack” consiste em fazer o servidor acreditar que o cliente não suporta uma versão recente/atualizada do TLS. Como é um ataque do tipo mitm, obriga a que haja acesso

ao meio onde a conexão se realiza, por exemplo, o router onde o cliente se liga. Assim que o cliente A envia pacotes ARP, um nodo malicioso pode adulterar valores e criar uma ARP cache à medida, interceptando a partir daí todos os pacotes entre o cliente e o servidor. Uma vez que os pacotes estão encriptados e a integridade é também garantida, não há forma de alterar a informação por forma a negociar a implementação de um protocolo SSL que utilize cifras inseguras. No entanto, impedir alguns pacotes de chegar ao cliente A e portanto não permitindo resposta ao servidor, este último interpreta como se não houvesse compatibilidade de protocolos e tenta uma nova negociação utilizando um protocolo mais antigo, permitindo assim explorar vulnerabilidades nas tais cifras inseguras.

3. Protocolo SSH

Pergunta P3.1

A primeira empresa escolhida foi a OVH, sendo o resultado do ssh-audit o seguinte:

```
user@CSI:~/Tools/ssh-audit$ python ssh-audit.py dinesys.es
# general
(gen) banner: SSH-2.0-OpenSSH_6.7p1 Ubuntu-5ubuntu1.4
(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 al
gorithm
                                            `- [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.
```

```
# 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
```

O software utilizado é o OpenSSH 6.7p1 com a versão SSH-2.0.

A segunda empresa escolhida foi a Outscale SASU, com o seguinte resultado do ssh-audit:

```
[exception] [Errno -2] Name or service not known
user@CSI:~/Aulas/tools/ssh-audit$ python ssh-audit.py 171.33.81.222
# general
(gen) banner: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8
(gen) software: OpenSSH 6.6.1p1
(gen) compatibility: OpenSSH 6.5+, 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-nistp521                    -- [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-nistp256                    -- [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

# host-key algorithms
(key) ssh-ed25519                           -- [info] available since OpenSSH 6.5
(key) ssh-rsa                               -- [info] available since OpenSSH 2.5.0, Dropbear SSH 0.28
(key) ecdsa-sha2-nistp521                   -- [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

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

```
# message authentication code algorithms
(mac) hmac-sha2-512-etm@openssh.com          -- [info] available since OpenSSH 6.2
(mac) hmac-sha2-256-etm@openssh.com          -- [info] available since OpenSSH 6.2
(mac) umac-128-etm@openssh.com               -- [info] available since OpenSSH 6.2
(mac) hmac-sha2-512                          -- [warn] using encrypt-and-MAC mode
                                             `- [info] available since OpenSSH 5.9, Dropbear SSH 2013.56
(mac) hmac-sha2-256                          -- [warn] using encrypt-and-MAC mode
                                             `- [info] available since OpenSSH 5.9, Dropbear SSH 2013.56
(mac) umac-128@openssh.com                   -- [warn] using encrypt-and-MAC mode
                                             `- [info] available since OpenSSH 6.2

# algorithm recommendations (for OpenSSH 6.6.1)
(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) -ecdsa-sha2-nistp521                   -- 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
```

Neste servidor, o software utilizado é o OpenSSH 6.6 1p1 com a versão SSH 2.0.
A versão mais vulnerável é a OpenSSH 6.6 1p1, sendo a vulnerabilidade mais grave a CVE-2016-8858, também para esta versão.
Apesar de apresentar uma elevada pontuação que supostamente corresponde à dimensão da gravidade, um relatório anexado na página da CVE não considera esta vulnerabilidade preocupante. Esta vulnerabilidade tem como consequência um DOS derivado de consumo de memória.