

Rede Overlay de Anonimização do Originador

Luis Pereira^{1[2]}, Carlos Afonso^{1[3]},

Gonçalo Nogueira^{1[4]}

¹ Universidade do Minho, Portugal

² a77667@alunos.uminho.pt, ³ a82529@alunos.uminho.pt, ⁴ a86617@alunos.uminho.pt

Abstração. Trabalho realizado na cadeira de Comunicações por Computador sendo a linguagem escolhida Java com o objetivo de aprofundar conhecimentos e técnicas de criação e configuração de redes, especificamente neste trabalho, utilizando uma topologia CORE específica e proceder ao seu anonimato. Para tal foram representados um cliente(Origin), um servidor(TargetServer) e uma rede Overlay Anon, formada pelo conjunto de gateways de transporte(AnonGW). Sendo assim criada uma rede em que Origin conecta-se por TCP porta 80, igualmente para o TargetServer, conectados por um túnel anonimizador UDP, a rede AnonGW. Sendo formada por gateways UDP de porta 6666 que procedem ao anonimato do Origin, passando por mais que um gateway, entregando ordenadamente os pacotes, encriptando os conteúdos de várias conexões TCP recebidas, conhecendo a lista de pares de outros gateways com quais pode estabelecer túneis e dos quais pode receber PDUs.

Keywords: UDP, GateWay, Anonimato, TCP, Servidor, Rede

1. Introdução

Privacidade e segurança online, um tema cada vez mais discutido atualmente devido ao crescimento exponencial do valor de dados pessoais valorizados pelas empresas mundiais, com esta corrida a recolhimento de dados pessoais, vários problemas de privacidade e segurança são levantados à medida que perdemos posse dos nossos próprios dados. Estes que depois serão usados e manipulados para criar tanto perfis para produtos assim como também em situações de segurança perfis pessoais de criminalidade, estando a par de existirem certas keywords que várias empresas sociais recolherem e notificam a forças policiais. Sendo cada vez mais necessário uma opção de escolha pelo utilizador de prosseguir ao seu anonimato e privacidade pessoal. O que tem levado recentemente ao enorme crescimento de serviços de VPN(Virtual Private Network). Um serviço mais avançado mas relativo a este trabalho efetuado.

2. Arquitetura da Solução

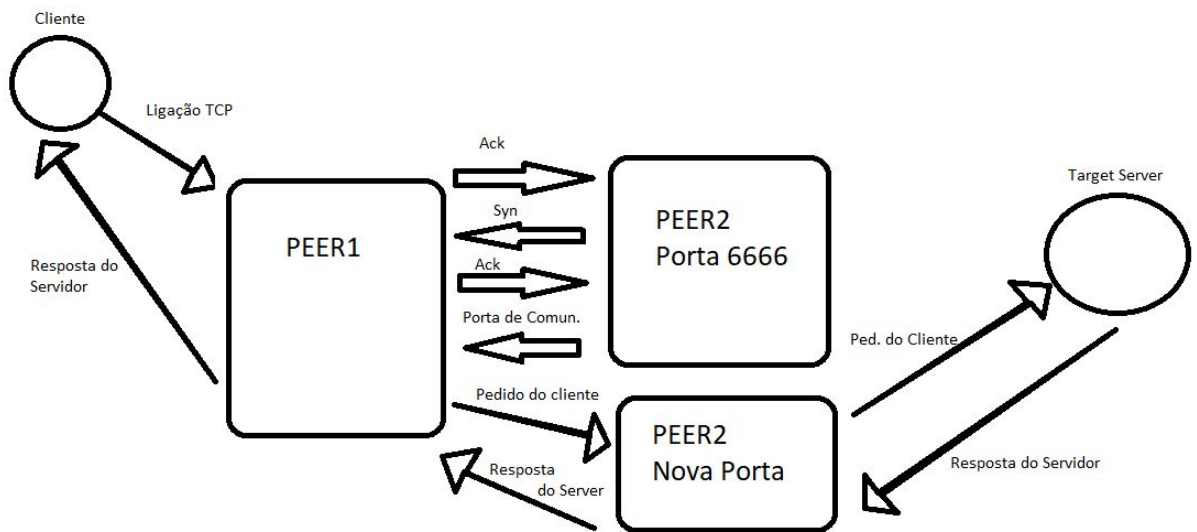


Fig. 1. A arquitetura estrutural da especificação do programa.

3. Especificacao do Protocolo UDP

Inicialmente, por parte do Peer1, é feita uma seleção aleatória do Peer destino a qual se irá conectar, efetuando um handshake. Durante este processo, serão trocadas mensagens entre os dois Peers, que são convertidas em bytes, mais concretamente, num array de bytes.

Numa segunda fase, quando o handshake é concluído com sucesso, Peer 1, recebe por parte do Peer 2 uma nova porta onde se envia o pedido do cliente, e quando possível, a resposta do servidor. Neste segundo processo, o array de bytes é alterado, devido à necessidade de segurança. Devido aos requisitos do método de encriptação, o array de bytes, para além de enviar o pedido e a resposta encriptada, também envia o tamanho dos dados encriptados.

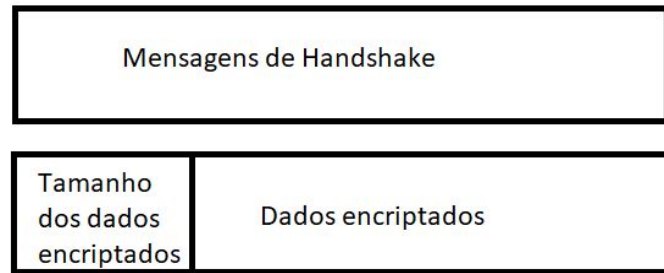


Fig. 2. Buffer correspondente às mensagens de Handshake e buffer correspondente à troca de pedidos e dados, correspondentemente.

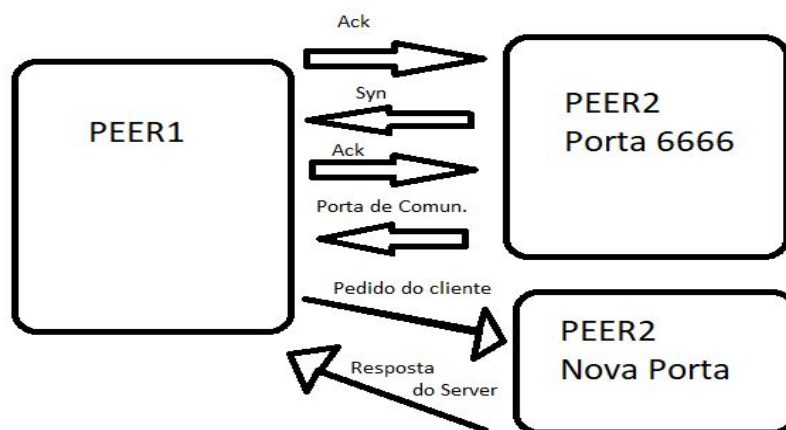


Fig. 3. Interações entre Peers

Tomámos a decisão de criar um protocolo de handshake, para garantir de alguma forma, a estabilidade que o TCP garante a nível de conexão entre os dois lados (Peer 1 e Peer 2).

Quando o Peer 1 recebe um pedido de conexão, após todos os procedimentos iniciais de TCP, serem concluídos, o Peer 1 envia um ACK para o Peer 2, assim que recebe este ACK, envia um SYN de volta para o Peer 1, caso este chegue com

sucesso, o Peer 1 envia de volta um ACK. Finalmente, o Peer 2 lança uma porta nova para escutar o pedido do Peer 1, e envia especificação dessa porta para o Peer 1.

A partir deste momento, o Peer 1 recebe o pedido do cliente, encripta a mensagem, e envia para o Peer 2, pela nova porta. Assim que o Peer 2 receber o packet, descripta e envia para o servidor. Quando o peer 2 receber a resposta do servidor, encripta-a e envia para o peer 1. Por fim, o peer 1, descripta, e envia de volta para o cliente.

4. Implementação

Este projeto foi implementado em JAVA, pois era a linguagem com que os elementos de grupo têm mais familiaridade em trabalhar.

Foi criada uma classe cuja única função era a de encriptação e decriptação, denominada de AESencrp. Aqui encontram-se os metodos referentes acima. Foi também criada a classe UDPtemp, que se encarrega de tratar da nova porta que é aberta pelo Peer 2, responsável pela passagem dos dados relativos ao pedido do cliente, e resposta do servidor. É com esta que se assegura a multiplexagem, em que vários clientes possam fazer pedidos ao mesmo tempo, pois o fluxo de dados será sempre feito por portas diferentes, garantido a separação dos fluxos.

A classe UDPworker tem como função a escuta de cada Anon que esteja ativo, pela porta 6666. É aqui que, quando o Peer 1 estabelece uma conexão com o Peer 2, é lançada a trama correspondente ao UDPtemp.

Foi também criada a classe AnonGW_Worker, que corresponde ao Peer 1 quando é “contactado” pelo cliente. Também é aqui que se recebe os packets do Peer 2, referentes à resposta do servidor, e posteriormente, enviados para o cliente.

Por fim, existe a classe anonGW, que é onde é feita a inicialização de cada anonGW, ou seja, sempre que um anon é iniciado, com os parâmetros corretos, é nesta fase, que a porta 6666 é lançada para escutar, e quando for recebido uma conexão do cliente, é lançada trama AnonGW_Worker.

A nível de bibliotecas usadas, para além das bibliotecas mais gerais, tais como, java.util, java.net, java.nio, entre outras, foi também usada javax.crypto e java.security, que dizem respeito à classe de segurança e encriptação.

5. Testes e Resultados

```

DEPOIS DE ENCRYPTAR= 00$H000400h00I/9Y050U!00g
                                F02W0!0G000%+AwJk0000!x}}00Cih00
4000B!0^0)00e0050000
                                8 0!z0oP0z00t0F03M00(0000=,5[0Ei000U0)CE0001E000000M0000?
00$0I00pK8[0000{00<00
                                010=F0000005
                                ;0`!00

```

Fig. 3. Dados após encriptação

```

root@Portatil1:/tmp/pycore.38948/Portatil1.conf
root@Portatil1:/tmp/pycore.38948/Portatil1.conf# wget 10.4.4.3/file1
--2020-05-24 17:02:44-- http://10.4.4.3/file1
Connecting to 10.4.4.3:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 193 [text/plain]
Saving to: 'file1.14'

100%[=====>] 193      --.-K/s   in 0.002s

2020-05-24 17:02:45 (82,5 KB/s) - 'file1.14' saved [193/193]

root@Portatil1:/tmp/pycore.38948/Portatil1.conf#

```

Fig. 4. Pedido efetuado e pacotes recebidos no Portatil 1

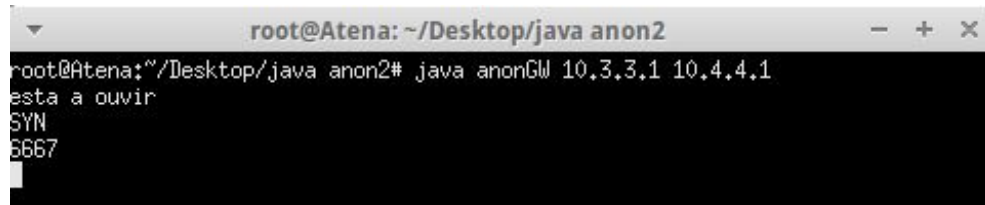
A terminal window titled 'root@Atena: ~/Desktop/java anon2'. The prompt is 'root@Atena:~/Desktop/java anon2#'. The user has entered the command 'java anonGW 10.3.3.1 10.4.4.1'. The output shows 'esta a ouvir' (listening), 'SYN', and '6667'.

Fig. 5. Escuta por parte do Peer 1


A terminal window titled 'root@Hermes: ~/Desktop/java anon2'. The prompt is 'root@Hermes:~/Desktop/java anon2#'. The user has entered the command 'java anonGW 10.3.3.1 10.4.4.3'. The output shows 'esta a ouvir' (listening), 'ACK', 'ACK', 'Pacote enviado' (packet sent), and 'Pacote enviado' (packet sent).

Fig. 6. Escuta por parte do Peer 2 e passagem de pacotes

6. Conclusões e trabalhos futuros

Este trabalho levou a uma maior apreciação pelo grupo pela segurança e privacidade pessoal de cada um e educando tanto sobre soluções assim como os vários obstáculos que tal implementação e capacidade de anonimato criam face à tecnologia correntemente utilizada para envio de pacotes. Sendo em uma rede normal, quase inexistente, não encriptadas e com informações sobre o IP do usuário e proveniente de internet da qual está a aceder.

O anonimato feito por encriptação e descentralização efetuando o seu envio por vários nodos random leva a um aumento de complexidade, maior overhead e aumento dos tempos de resposta, passando por Nodes em vez de ser entregue quase diretamente. Por tais razões e por outras, provenientes de internet muito provavelmente não usarão tais métodos, pelo que teremos de recorrer a outras empresas de VPN para termos acesso a tais opções, funcionando de maneira similar em que existe um Nodo intermediário pelo qual a nossa ligação é efetuada, no melhor dos casos o logs de tal conexão seriam apagados para maior anonimato, o que atualmente pode não ser o caso em alguns provenientes de tal serviço. Certamente o grupo está mentalmente melhor preparado para adaptar trabalhos futuros com bases e estratégias de anonimato.