

REDE OVERLAY DE ANONIMIZAÇÃO DO ORIGINADOR

Carolina Cunha, Hugo Faria, João Diogo Mota

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail:{a80142,a81283, a80791}@alunos.uminho.pt

INTRODUÇÃO

Hoje em dia, devido aos grandes avanços tecnológicos que têm vindo a ocorrer, cada vez mais a vida de cada um depende e está exposta online. Como tal, além de toda a panóplia de vantagens e facilidades que isto traz, também existem, naturalmente, desvantagens, sendo a questão da segurança/privacidade uma das principais. A privacidade é algo impossível de alcançar por completo quando se trata da troca de informação online. Uma das formas de minimizar os problemas de segurança passa pela tentativa de camuflar a verdadeira origem das conexões, tal como o que é proposto pela equipa docente na realização do projeto em causa.

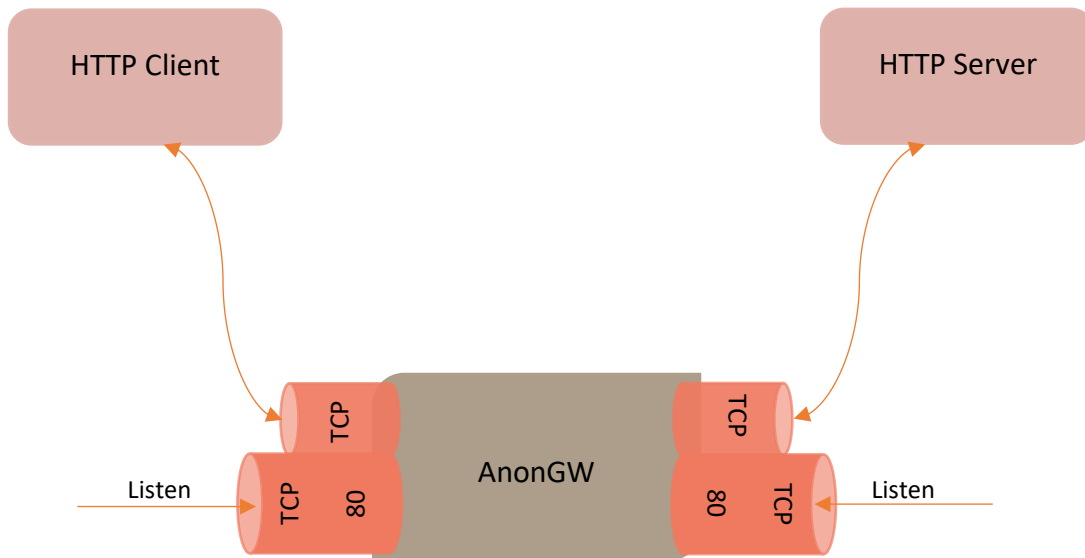
Com este projeto, pretende-se não conexão direta de um determinado cliente de origem a um Target Server através de uma conexão TCP, mas a utilização de uma Rede *Overlay* de Anonimização. Desta forma, será possível que o IP de origem do cliente não seja denunciado.

A Rede *Overlay* de Anonimização referida é constituída por Gateways de Transporte (AnonGW) por toda a rede. Quando o cliente pretende comunicar com o servidor, ao invés da sua conexão ser feita diretamente com o Target Server, esta será redirecionada para um dos AnonGW presentes na rede. Para acrescentar mais um nível de indireção, foi também pedida a passagem da conexão por um novo AnonGW. Esta conexão será, por sua vez, feita usando UDP, para uma maior eficiência e versatilidade, principais vantagens da utilização do protocolo UDP, tal como estudado. Por fim, o pedido ao Target Server será feito pelo segundo AnonGW.

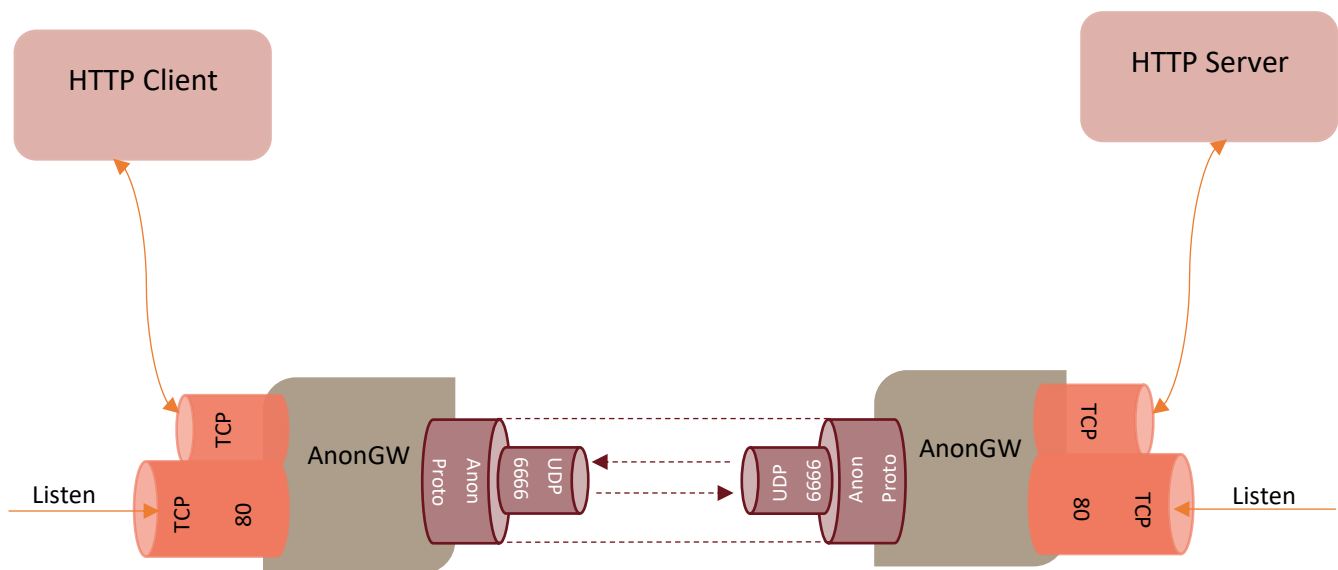
Para a realização deste projeto, foi eleita a linguagem de programação JAVA.

ARQUITETURA DA SOLUÇÃO

FASE I



FASE II



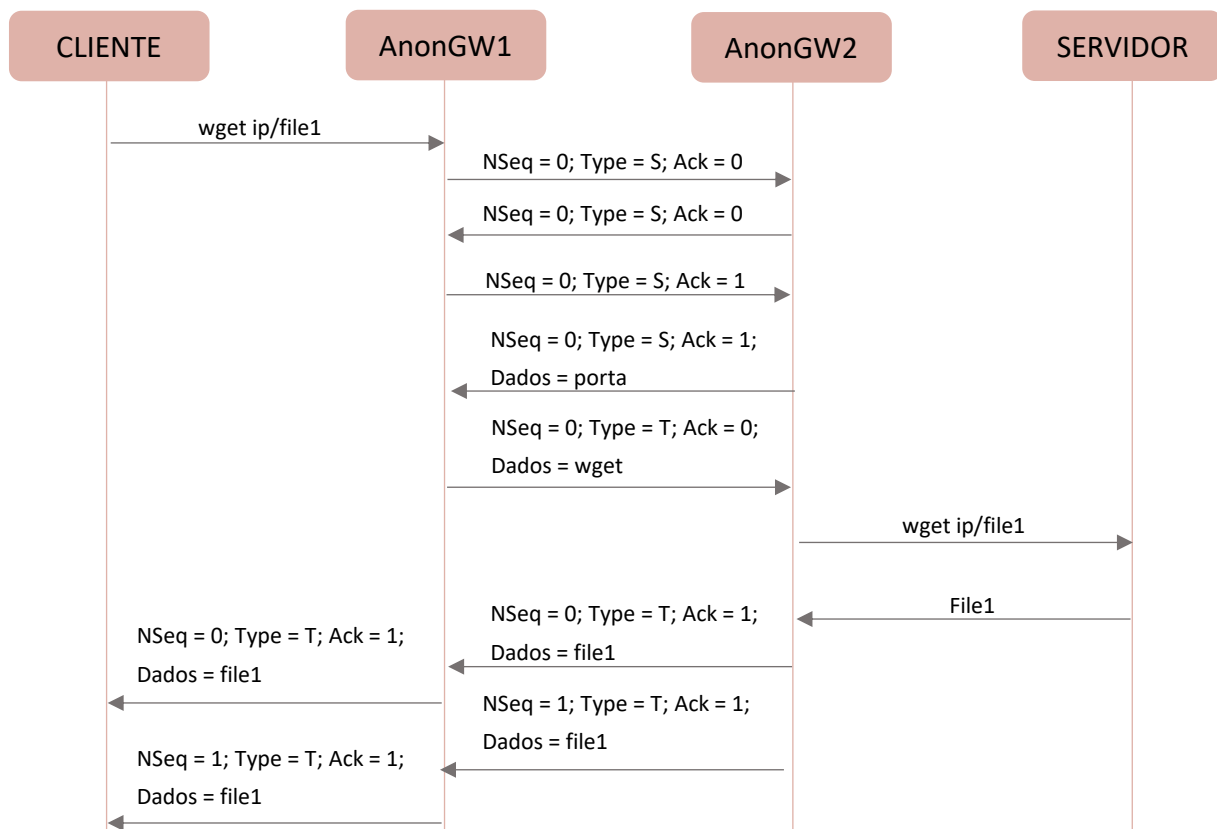
ESPECIFICAÇÃO DO PROTOCOLO UDP

FORMATO DAS MENSAGENS PROTOCOLARES (PDU)

O pacote de dados é formado pelos seguintes campos:

- **nSeq**: Número de sequência. Identifica o número do pacote que está a ser transferido;
- **Tipo**: Tipo de pacote. S, se se pretende estabelecer conexão; T, se se pretende realizar uma transferência de ficheiros;
- **Ack**: Diferenciação de pacotes do mesmo tipo;
 - S 0, para pedir ligação;
 - S 1, para pedir porta;
 - T 0, para pedir transferência;
 - T 1, para transferir.
- **Dados**: Os dados (*payload*) transferidos.

INTERAÇÕES



IMPLEMENTAÇÃO

BIBLIOTECAS DE SUPORTE

Para o desenvolvimento deste projeto, foram utilizadas bibliotecas relativas a *DatagramPacket*, *DatagramSocket*, *Socket*, *ServerSocket* por forma a ser possível formar os pacotes e interligar endereços IP de terminais. A biblioteca *Java Cryptography* permitiu a encriptação dos dados transmitidos entre os canais UDP.

FASE I

A realização do problema foi de origem dividido em duas fases. Numa primeira fase, foi desenvolvida a conexão TCP do cliente com um dos AnonGW, sendo que este, contrariamente à solução final, recebe a conexão TCP e liga-se diretamente Target Server por uma conexão TCP.

A classe AnonGW foi implementada com o objetivo de permanecer à escuta de novos pedidos de conexão. No momento em que recebe um pedido por parte de um cliente na porta 80 (porta em que está à escuta), este cria um *thread* para que o ClientHandler lide com os pedidos do cliente.

O ClientHandler irá, por sua vez, criar um *socket* TCP para a comunicação com o Target Server. Assim sendo, vão ser criadas duas *threads* que irão permitir a leitura e a escrita por parte do servidor, *serverReader* e *serverWriter*, respetivamente.

Do lado do servidor, e seguindo a mesma lógica, foram criadas duas classes. Uma delas, *Servidor*, vai estar à escuta de pedidos de conexão por parte de um AnonGW. Quando é recebido um pedido, esta cria uma *thread* para que o ServerHandler fique responsável pelos pedidos do servidor.

FASE II

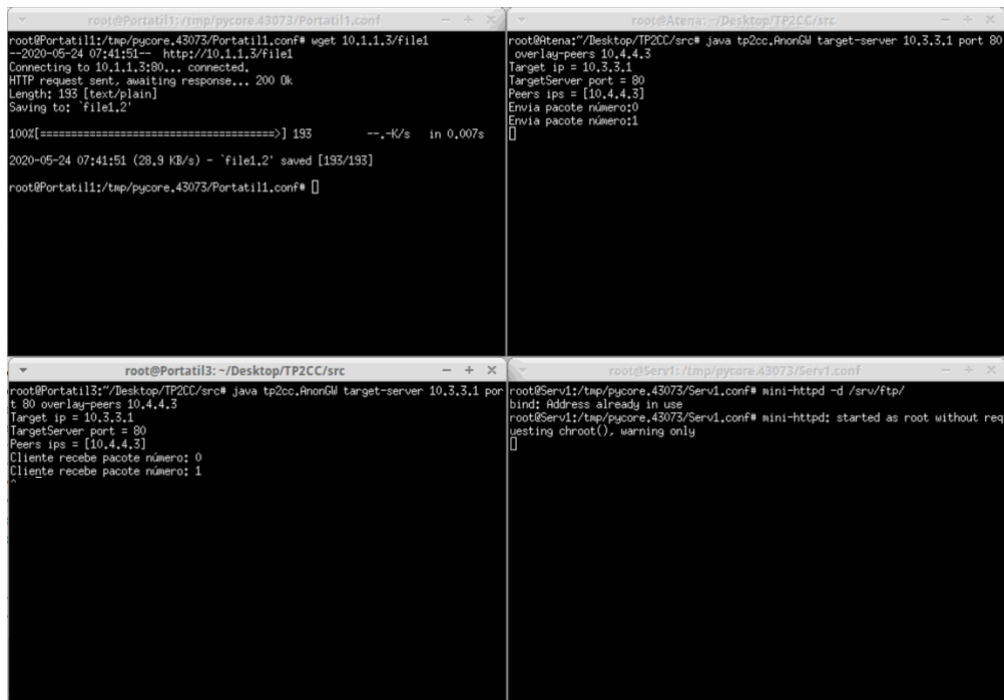
Na segunda fase do projeto, pretende-se o desenho e implementação do Anon Protocol sobre UDP, entre os dois AnonGW.

O AnonGW vai criar uma *thread* cliente e uma *thread* servidor, de modo a que qualquer AnonGW consiga fazer o papel de ler/escrever do cliente ou servidor. A *thread* do cliente (UDPclientworker) vai enviar um pacote para AnonGW do lado do cliente, permitindo que este saiba que existe um pedido de ligação. Caso o servidor não esteja ocupado com outro cliente na porta 6666, vai enviar uma mensagem de retorno ao cliente. Nesta mensagem, encontrar-se-á nos dados uma nova porta, possibilitando a troca de dados. Após receção do pacote, o cliente vai conectar-se a um dos *peerIPs* (AnonGW) através desta porta, iniciando o processo de envio de informação.

Por outro lado, a *thread* do servidor (UDPserverworker) estará à espera da receção de um pacote por parte de um AnonGW. Após receção, vai devolver uma nova porta e criar uma *thread* UDPextraserworker, possibilitando atribuir a esta *thread* a responsabilidade de transferência de dados. Enquanto isto, a *thread* UDPserverworker ficará à espera de um novo pedido de ligação. Após ligação da *thread* UDPextraserworker à porta atribuída, esta fica à espera de dados enviados pelo cliente. Assim que lê estes dados, vindos do UDPclientworker, reenvia-os para o servidor. Quando o servidor responder, os dados serão lidos e percorrerão o caminho inverso, até chegarem ao seu destino.

Durante todo o percurso entre os AnonGW, os dados dos pacotes são encriptados e posteriormente desencriptados, pelo que a transferência entre os *sockets* TCP são desencriptadas.

TESTES E RESULTADOS



```
root@Portatil1:/tmp/pycore.43073/Portatil1.conf# wget 10.1.1.3/file1
--2020-05-24 07:41:51-- http://10.1.1.3/file1
Connecting to 10.1.1.3:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 193 [text/plain]
Saving to: 'file1.2'

100%[=====] 193 --.-K/s in 0.007s

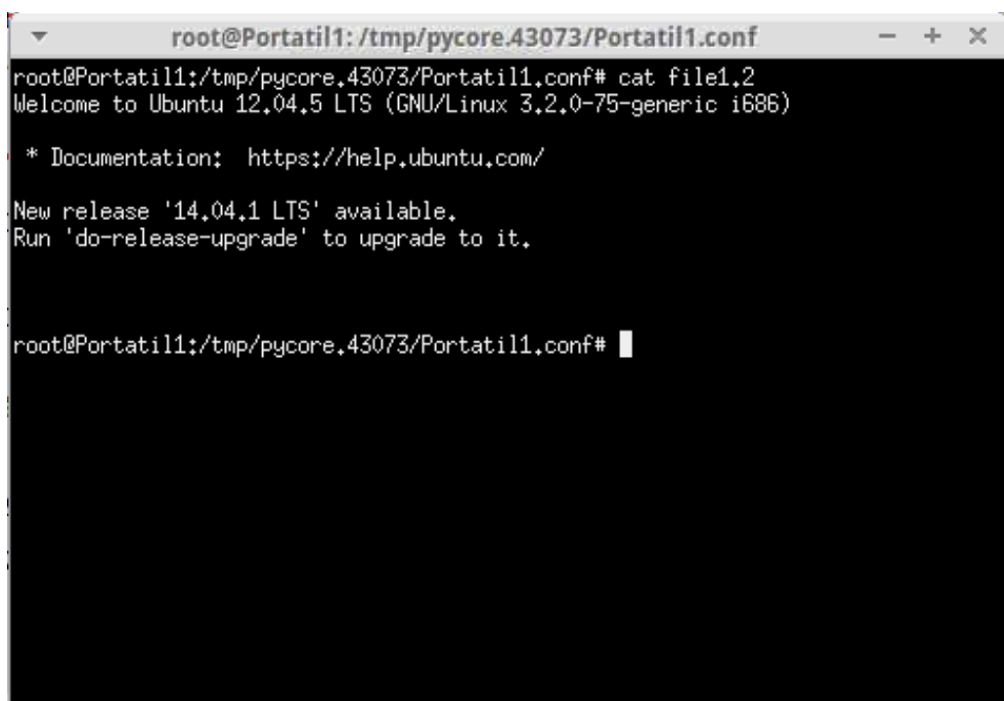
2020-05-24 07:41:51 (28.9 KB/s) - 'file1.2' saved [193/193]
root@Portatil1:/tmp/pycore.43073/Portatil1.conf#

root@Atena:~/Desktop/TP2CC/src# java tp2cc.AnoniGM target-server 10.3.3.1 port 80
overlay-peers 10.4.4.3
Target ip = 10.3.3.1
TargetServer port = 80
Peers ips = [10.4.4.3]
Envia pacote número:0
Envia pacote número:1
[]

root@Portatil3:~/Desktop/TP2CC/src# java tp2cc.AnoniGM target-server 10.3.3.1 port 80 overlay-peers 10.4.4.3
Target ip = 10.3.3.1
TargetServer port = 80
Peers ips = [10.4.4.3]
Cliente recebe pacote número: 0
Cliente recebe pacote número: 1
^

root@Serv1:/tmp/pycore.43073/Serv1.conf# mini-httpd -d /srv/ftp/
bind: Address already in use
root@Serv1:/tmp/pycore.43073/Serv1.conf# mini-httpd: started as root without req
uesting chroot(), warning only
[]
```

Figura 1: Teste transferência File1



```
root@Portatil1:/tmp/pycore.43073/Portatil1.conf# cat file1.2
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-75-generic i686)

 * Documentation:  https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

root@Portatil1:/tmp/pycore.43073/Portatil1.conf#
```

Figura 2: File1 transferido

Como é possível verificar pelas figuras anteriormente apresentadas, a transferência do *file1*, através de uma rede *overlay* de anonimização, foi realizada com sucesso.

De entre os requisitos mínimos obrigatórios, confirma-se a realização da encriptação dos dados (*payload*) dos pacotes através da biblioteca *Java Cryptography*, garantindo assim a cifragem do conteúdo para evitar espionagem.

No que diz respeito à entrega ordenada, (os dados recebidos de um extremo são entregues no outro extremo pela mesma ordem com que foram recebidos), esta conseguiu-se graças ao envio dos pacotes imediatamente após a sua receção.

Por fim, a multiplexagem de clientes (capacidade de lidar ao mesmo tempo com mais que uma conexão TCP, separando os fluxos convenientemente), foi garantida graças à existência de diferentes portas, atribuídas a cada cliente, permitindo o descongestionamento do servidor e dando possibilidade a vários clientes de comunicarem com o servidor em simultâneo.

CONCLUSÕES E TRABALHO FUTURO

A elaboração deste trabalho prático permitiu aprofundar os conhecimentos relativos aos protocolos de transporte TCP e UDP, bem como a necessidade e importância da existência da segurança, de modo a tentar proteger a privacidade pessoal de cada cidadão.

Ao longo da realização do projeto, o grupo deparou-se com questões tais como qual a melhor abordagem para conseguir camuflar a verdadeira origem das conexões, uma vez que é impossível alcançar a verdadeira noção de “privacidade”.

Desta forma, foram estudadas diversas formas de encriptação e de comunicação entre as diversas máquinas dispersas pela rede que, depois de discutidas com o docente da unidade curricular, iniciou-se a implementação das mesmas.

Assim, verificou-se através de capturas *Wireshark* a anonimização da Rede *Overlay*, permitindo confirmar a ligação segura entre o cliente e o servidor.

Numa retrospectiva, o grupo concluiu que a resolução do problema foi concluída com sucesso. No entanto, nem todos os objetivos pretendidos foram realizados. Esta dificuldade deveu-se a complicações no manuseamento da máquina virtual fornecida pelos docentes, em que os principais problemas passaram pelo facto de que esta deixou de funcionar para dois elementos do grupo e nenhum elemento do grupo conseguiu ter acesso à internet a partir da mesma, a dada altura da realização do trabalho. Por estes motivos, houve um grande atraso no avanço do projeto, que impediram um melhor proveito do projeto.

Apesar disso, todos os elementos do grupo consolidaram os conteúdos lecionados nas aulas teóricas na unidade curricular, conhecendo as vantagens do uso dos dois protocolos de transporte estudados, e conseguindo aplicá-las na prática, permitindo assim proteger o *Target Server* e manter uma sessão confidencial para troca de dados entre o cliente e o servidor.

Numa perspetiva de trabalho futuro, e assumindo que quaisquer problemas relacionados com as máquinas virtuais estariam resolvidos, o grupo gostaria de implementar o controlo de perdas e autenticação por assinatura digital de cada PDU, aperfeiçoando a segurança, estabilidade e completude da comunicação.

