CSD Project 2 - Report

Rafael Costa - 60441
Francisco Silveira - 60816

Inicialmente, são carregados os ficheiros dos dados dos nós TOR, bem como o Cliente Input. Posteriormente, para cada nó TOR e para o cliente e destino, é identificado o país de origem do endereço IP, através da biblioteca geoip.

O procedimento subsequente consiste na seleção do circuito. Inicialmente, são selecionados os nós que podem ser "exits". Cada nó possui um atributo "exit", no qual são especificados os endereços de destino que são ou não aceites. Esta informação é tida em consideração durante a seleção dos nós que podem ser "exits".

Posteriormente, procede-se à seleção dos guards, que podem corresponder a qualquer nó.

Em conformidade com o procedimento adotado na rede TOR, foi realizada uma seleção sequencial dos nós, iniciando pelo nó de guard, seguido pelo nó de exit e, por fim, pelo nó de middle. Consideramos esta a opção mais intuitiva e segura, uma vez que possibilita a utilização das informações contidas no nó precedente para a seleção do nó subsequente, garantindo, deste modo, a continuidade do percurso através de uma sequência de nós segura.

A seleção de qualquer nó é efetuada através da utilização de um atributo trust, o qual sofre uma diminuição caso se concretizem determinadas condições.

Para cada guarda possível, procede-se à verificação da correspondência entre o guarda e o cliente no âmbito da mesma aliança. Em caso afirmativo, a confiança que o cliente deposita nessa aliança é atribuída. Na ausência de uma aliança que abranja os dois países, é atribuído um nível de confiança padrão de 0,5.

Em última análise, é necessário verificar se os dois nós estão localizados no mesmo país. Se tal for o caso, a confiança deve ser multiplicada por 0,5, diminuindo a mesma.

Posteriormente, o nó de guarda é retirado da enumeração de nós de saída, com o propósito de excluir de forma definitiva a possibilidade de o mesmo nó surgir mais de uma vez no circuito.

No caso de o exit pertencer à mesma família que o guard, a trust nesse nó de exit é reduzida para 10%. Se pertencer à mesma ASN, a trust é reduzida para 20%. Se estiver no mesmo país que o cliente, a trust é reduzida para 50%. Se tiver no mesmo país que o destino, a trust é reduzida para 50%. Se estiver no mesmo país que o guard, a trust é reduzida para 50%. Estas reduções de trust são acumulativas, de modo a tentar garantir que esse nó não deve ser escolhido. A trust final daquele nó é guardada num atributo trust do próprio nó.

Posteriormente, os nós de guard e de exit selecionados são removidos da lista de nós de middle, com o objetivo de eliminar definitivamente a possibilidade de o middle ser um nó repetido no circuito.

A seleção do middle é, por sua vez, aleatória, embora controlada. Procedeu-se à seleção aleatória de um nó middle, tendo sido subsequentemente verificada a não coincidência do mesmo com outro nó ou família. No caso de tal coincidência, o processo de seleção aleatória e verificação será repetido até ser encontrado um nó que não apresente o mesmo ANS e/ou família que os restantes. Importa salientar que este processo só pode ser repetido até 15 vezes por motivos de eficiência. Se, após a última iteração, as condições ideais não tiverem sido alcançadas, o último nó selecionado é o nó retornado. Assim permitimos uma seleção de middle mais eficiente.

O limite de 15 tentativas foi escolhido por se considerar um valor equilibrado: suficientemente elevado para garantir uma boa probabilidade de encontrar um nó que cumpra os critérios, mas suficientemente baixo para não comprometer a eficiência do processo. Dado o tamanho e diversidade da rede Tor, a probabilidade de não se encontrar um nó adequado dentro dessas 15 tentativas é reduzida, tornando esta abordagem eficaz.

Posteriormente, os nós de guard e exit são sujeitos à função categorize_and_select.

A função categorize_and_select é responsável por escolher um nó (guard ou exit) com base na sua trust e largura de banda. Primeiro, ordena os nós por confiança decrescente e tenta selecionar um subconjunto com trust acima de um limiar "seguro" (safe_upper) cuja largura de banda acumulada atinja uma fração mínima da largura de banda total (bandwidth_frac). Se não for possível, tenta novamente com nós "aceitáveis", usando um limiar mais baixo (accept_upper). Se ainda assim não houver candidatos suficientes, a função falha, e o sistema recorre ao melhor relay disponível como alternativa.

Concluindo, decidimos aplicar um foco mais a segurança e privacidade pois achamos que é a principal razão por alguém escolher a rede TOR. Sabemos também que a eficiência é bastante importante, porém mesmo termos focado mais na segurança tentamos escrever o código da forma mais eficiente possível e sem percorrer listas desnecessariamente. Para além disto o código é extremamente rápido a procurar o melhor caminho demorando entre 0.5 a 1.1 segundos.