Detecção de Usuários *Smurf* na Rede da Steam

Caio C. R. Ramos[1], Christian Cardozo[2]

[1]Escola Politécnica – Universidade Federal do Rio de Janeiro (UFRJ)
Caixa Postal 68.529 – 21941-909 – Rio de Janeiro – RJ – Brasil

[2]Programa de Engenharia de Sistemas e Computação - PESC/Coppe/UFRJ
Caixa postal: 68.511 - 21941-972 – Rio de Janeiro - RJ – Brasil

caiocrr@poli.ufrj.br, christiancardozo@cos.ufrj.br

Abstract. *This article describes a potential solution for the Smurfs - fake accounts - problem in Steam social network. Smurfs are secondary accounts from users that have a higher level main account. This account type causes unfair matches in competitive games and currently there is no deterministic solution. In this work, we introduce a probabilistic approach for Smurfs detection using a Steam subnetwork snapshot.*

Resumo. *Este meta-artigo descreve uma possível solução para a detecção de usuários Smurfs na rede de relacionamento da Steam. Usuários Smurfs são contas secundárias de jogadores que possuem um nível diferente de sua conta principal. Esse tipo de conta causa desbalanceamento em jogos competitivos e atualmente não existe uma solução determinística para o problema. Introduziremos uma solução probabilística para a detecção de usuários Smurfs a partir de uma sub-rede de relacionamento da Steam.*

1. Introdução

A quantidade de contas falsas em todo tipo de rede social aumentam proporcionalmente com o número de usuários reais ativos da rede [1]. Estas contas existem para diversos propósitos, dependendo do contexto da rede. Na rede da Steam [2], jogadores criam contas *Smurfs* com o objetivo de jogar em um nível competitivo diferente daquele que sua conta principal está. Isso causa uma grande turbulência no cenário de jogos competitivos como um todo, tanto nos níveis mais altos quanto nos mais baixos.

Detectar uma conta *Smurf* não é uma tarefa trivial. Atualmente, na rede Steam, as contas são verificadas manualmente através do seguinte processo: usuários reportam as contas suspeitas e as contas com maiores ocorrências de denúncias serão fortemente verificadas por algum empregado da Steam para, enfim, uma atitude ser tomada. Pretendemos desenvolver uma solução que facilite a detecção de prováveis contas *Smurfs* através de um ranqueamento das contas, onde contas com baixo ranqueamento tenham maior probabilidade de serem contas falsas.

2. Trabalhos relacionados

Existem diversos trabalhos relacionados a detecção de contas falsas em variadas redes sociais, podendo cada uma possuir características topológicas diferentes. Em redes

sociais largamente utilizadas, como Facebook [3] ou Twitter [4], contas falsas podem existir tanto para algum tipo de fraude quanto para se passar por outra pessoa virtualmente. Mesmo as redes sendo diferentes umas das outras e as contas falsas possuírem objetivos diversos, podemos usar a estrutura das redes para facilitar a busca por *Smurfs*. Os trabalhos [1], [5] e [6] utilizam desta abordagem para conseguir formular métricas de ranqueamento com o objetivo de detectar usuários *Smurfs*.

3. Proposta

Seguindo a abordagem proposta por Cao [1], podemos dividir uma rede social em duas partes: uma sub-rede de contas confiáveis e outra de contas não confiáveis (*Smurfs*). A estrutura da rede tem papel fundamental na definição desta divisão. A ideia é baseada na hipótese de que contas *Smurfs* tendem a ficarem próximas na rede, podendo inclusive formar diversas comunidades. Com isso, existiria um número limitado de arestas que ligariam a sub-rede de contas confiáveis à essas diversas comunidades onde há contas *Smurfs* (Figura 1). Para identificarmos potenciais *Smurfs*, podemos realizar uma Random Walk (RW) com um número de passos limitados, como descrito em [1], onde escolhemos alguns nós da rede como sendo confiáveis e atribuímos à eles uma confiança inicial. O resultado esperado neste caso é que interrompendo a RW antes do ponto de convergência, os vértices *Smurfs* não receberiam a confiança propagada pela rede.

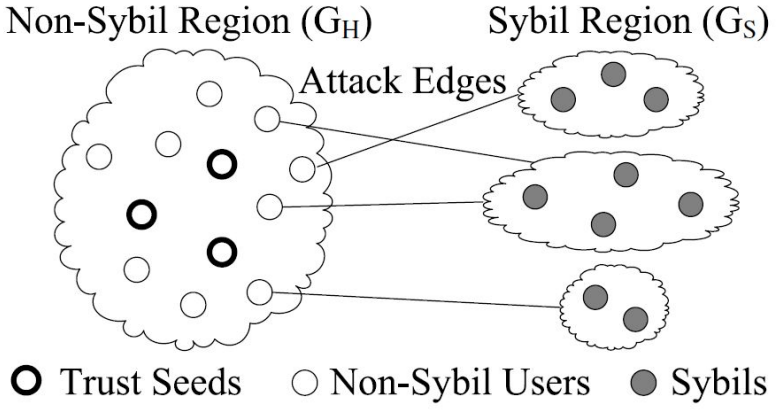

Figura 1. Estrutura da rede social. Sybil region é a região que contém vértices não confiáveis; Non-Sybil Region contém vértices confiáveis. Fonte: [1].

Para avaliar esta proposta, foi utilizado um conjunto de usuários reduzido da Steam, contendo 114047 usuários. Este conjunto de usuários foi gentilmente fornecido pelo Prof. Daniel R. Figueiredo (PESC/Coppe/UFRJ) e por seu aluno Gabriel Sab (Escola Politécnica - UFRJ). Utilizando esta base de dados, foi montada uma rede com todos os usuários, de onde foram extraídas sub-redes induzidas por jogo jogado. Para as avaliações, foram escolhidas as sub-redes dos jogos Counter-Strike: Global Offensive (CS:GO) [7] e Path of Exile (PoE) [8]. Não há uma forma determinística consolidada para a identificação de um usuário *Smurf*, então, o foco passa a ser exatamente conseguir filtrar usuários com grandes suspeitas na rede para posteriormente passarem por uma verificação manual (Figura 2).

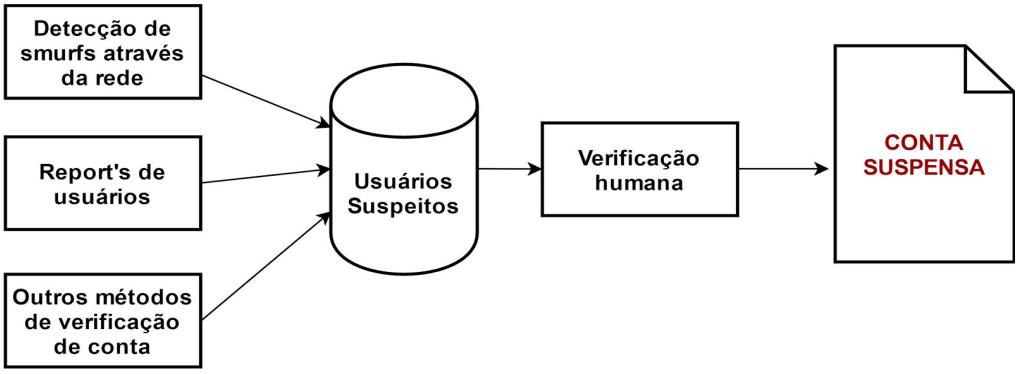

Figura 2. Processo para a suspensão de uma conta *Smurf*. A análise da conta por redes é uma forma probabilística para detectar usuários suspeitos.

3.1 Estrutura da rede

Nesta seção, as características estruturais da rede original e das duas sub-redes serão avaliadas. A rede inicial da Steam, com todos os vértices, possui 114047 vértices, 2197420 arestas e é uma rede conexa. A rede do CS:GO contém 77276 vértices, 1085 componentes conexas, onde 76119 vértices estão numa componente conexa gigante. A rede do PoE contém 23117 vértices, 2893 componentes conexas, onde 19893 vértices estão numa componente conexa gigante. Como as duas sub-redes possuem diversas componentes conexas de ordem muito menor comparada à maior componente, a partir deste ponto, apenas a maior componente conexa de cada rede será analisada e utilizada nos algoritmos. Na Tabela 1 estão especificadas outras características das redes (levando em consideração apenas a maior componente conexa de cada). A Tabela 2 especifica dados sobre os graus de cada rede. No Anexo A, podem ser encontrados os gráficos das CCDF empíricas de grau das três redes em escala log-log.

Tabela 1. Características gerais das redes Steam, CS:GO e PoE

	Vértices	Arestas	Diâmetro	Densidade	Clusterização
Steam	114047	2197420	6	0.0003378	0.073329286
CS:GO	76119	1457030	16	0.0002514	0.065913889
PoE	19893	148484	17	0.0003752	0.100848954

Tabela 2. Características dos graus das redes Steam, CS:GO e PoE

	Grau médio / Desvio Padrão	Grau máximo	Grau mínimo
Steam	38.5353 / 46.4548	966	2
CS:GO	19.1392 / 21.0035	381	1
PoE	7.4244 / 8.1380	125	1

3.2 Algoritmo de Ranqueamento

O algoritmo utilizado para identificar contas *Smurfs* é baseado no SybilRank proposto por Cao [1]. O primeiro passo do algoritmo é definir vértices na rede que garantidamente são contas reais. Então, se define uma mesma confiança inicial maior que zero para cada um destes vértices confiáveis. Os outros vértices da rede são inicializados com confiança igual à zero. O segundo passo do algoritmo é executar o algoritmo de RW na rede. Nesse caso, o RW clássico irá propagar a confiança de acordo com o grau de cada vértice, obedecendo a seguinte equação:

$$(1)\ x(i) = \sum_{j \in vizinhos(i)} x(j)/deg(j)$$

A RW, ao atingir o estado estacionário, terá definido o valor de cada vértice sendo proporcional ao grau do mesmo, dada pela seguinte equação:

$$(2)\ \pi(i) = deg(i)/2m$$

Para que tenhamos um ranqueamento confiável, devemos finalizar a RW antes do estado estacionário. Cao [1] define que são necessários log(n) iterações da RW, onde n é o número de vértices da rede. A Figura 3 demonstra uma RW numa rede depois de quatro iterações e no seu estado estacionário. Com log(n) passos da RW, os vértices com maior probabilidade de serem *Smurfs* terão um valor baixo no ranqueamento, e os com maior probabilidade de serem contas reais terão valores mais elevados. Na seção 3.4 será definido uma forma de propagar confiança de acordo com atributos da rede, buscando melhorar as estimativas do algoritmo para as sub-redes da Steam.

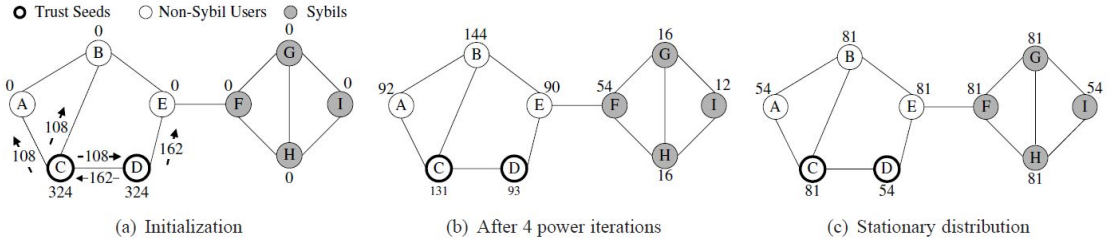

Figura 3. Confiança de cada vértice durante iterações da propagação de confiança. Fonte: [1].

3.3 Determinação de Vértices Confiáveis

O algoritmo descrito na seção 3.2 espera receber uma lista de vértices confiáveis. Precisamos selecionar uma quantidade limitada de vértices na rede que inicialmente serão responsáveis por propagar confiança. Sendo c o número de vértices em que selecionamos como confiáveis, a confiança individual inicial para cada vértice i será:

$$(3)\ T(i) = 100/c$$

Os demais vértices da rede serão inicializados com confiança igual à zero. Neste momento, nos deparamos com a tarefa de selecionar vértices confiáveis. Para essa identificação, primeiro aplicamos o algoritmo Louvain Method [9] que identifica a formação de comunidades na rede. Para cada comunidade, os vértices são ordenados de

acordo com quantidade de horas jogadas, um atributo presente para cada usuário na base de dados da Steam. O 5 primeiros de cada comunidade foram analisados manualmente e aqueles que, empiricamente, tinham maiores garantias de serem contas reais, foram selecionados. Foram identificadas 49 comunidades na sub-rede CS:GO, e 231 vértices foram selecionados como confiáveis seguindo a análise empírica. Para a sub-rede PoE, foram identificadas 55 comunidades e 269 vértices foram selecionados como confiáveis.

3.4 Propagação de confiança

Normalmente, as rede sociais possuem poucas *attack edges* (arestas que interligam contas reais à contas *Smurfs*) comparado à quantidade total de arestas [1]. Portanto, um usuário confiável dificilmente terá um usuário *Smurf* em sua lista de amigos. Mas, se esse tiver, não devemos propagar a confiança desse usuário para o usuário *Smurf* da mesma forma que devemos propagar para um vizinho não *Smurf*. A fim de evitar este problema, utilizamos um índice de propagação de confiança dado pela seguinte equação:

$$(4)\ B(i,j) = \textit{tempo de amizade}(i,j)/\textit{tempo de conta criada}(i)$$

Ou seja, o vértice i irá propagar mais confiança para usuários que possuem maior tempo de amizade, normalizando pelo tempo de conta criada. Ambos atributos estão presentes na base de dados da Steam. Essa normalização foi definida baseada na verificação empírica de que usuários *Smurfs*, muito dificilmente, terão relacionamento de longa data com usuários confiáveis na rede. Neste momento, as redes CS:GO e PoE passam a ser direcionadas, onde cada aresta não-direcionada dá origem a duas arestas direcionadas com o peso igual ao índice de normalização descrito em (4).

4. Resultados

O algoritmo proposto foi exatamente executado como descrito na seção 3 para as sub-redes CS:GO e PoE. A tabela 3 contém os parâmetros de confiança inicial e números de iterações da RW para cada sub-rede.

Tabela 3. Valores iniciais de confiança e número de iterações da RW nas sub-redes.

	Vértices Confiáveis	Confiança Inicial	Somatório das Confianças	Número de Iterações
CS:GO	231	0.432900433	100	5
PoE	269	0.371747212	100	4

A Tabela 4 contém, os cinco primeiros e cinco últimos valores de ranqueamento para as redes CS:GO e PoE. No Anexo B podem ser encontradas as CCDF empíricas dos valores de ranks retornados por cada sub-rede em escala log-log.

Tabela 4. Maiores e menores valores de rank nas sub-redes

CS:GO		PoE	
5 maiores ranks	5 menores ranks	5 maiores ranks	5 menores ranks
1.19557068539338	0.0	0.70174095379021	0.0
0.9836515193467	0.0	0.58823794911660	0.0
0.76737029868827	0.0	0.46523810318386	0.0
0.53787410553390	0.0	0.45705513708051	0.0
0.49143524916933	0.0	0.44942123546996	0.0

As duas sub-redes possuem 15593 usuários em comum. Foi feita uma análise sobre esses vértices em comum seguindo a seguinte ideia: particionar o ranking em *b* partes iguais, chamadas de bins, e para cada bin *i*, verificar a proporção de vértices em comum comparado ao bin *i* da outra rede. Esta abordagem foi aplicada para dois bins e para quatro bins (Figura 4). Com dois bins, temos uma acurácia acumulada de 0.71, e para quatro bins, acurácia acumulada de 0.44. Esta acurácia representa a similaridade entre os bins de mesmo índice levando em consideração os 15593 vértices em comum.

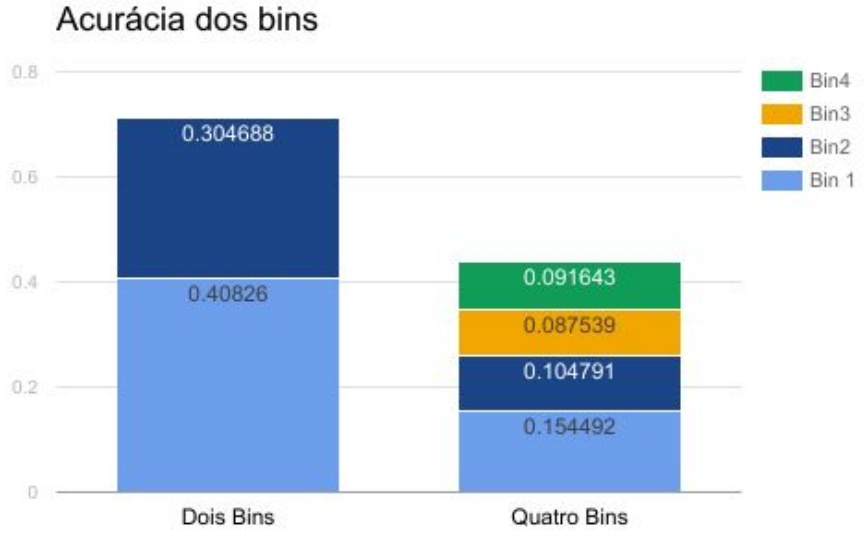

Figura 4. Acurácia dos bins em relação à interseção entre eles para os 15593 vértices.

Como o ranking nos fornece uma solução proporcionalmente probabilística, foi realizada uma verificação manual das contas de baixo ranqueamento para determinar se realmente são *Smurfs*. A heurística de verificação consistiu em analisar o nível da conta, horas de jogo, conquistas obtidas, lista de amigos, entre outras informações presentes no site da Steam. Para a sub-rede PoE, foram escolhidos aleatoriamente 100 dos últimos

1000 vértices do ranking para análise. Dos analisados, 65 contas foram classificadas empiricamente como *Smurfs*. Na sub-rede CS:GO, foram escolhidos aleatoriamente 150 dos últimos 2000 vértices do ranking. Destas, 102 contas foram classificadas como *Smurfs*. Temos na Figura 5, a porcentagem de contas corretamente classificadas com baixa confiança nos rankings seguindo a verificação manual feita.

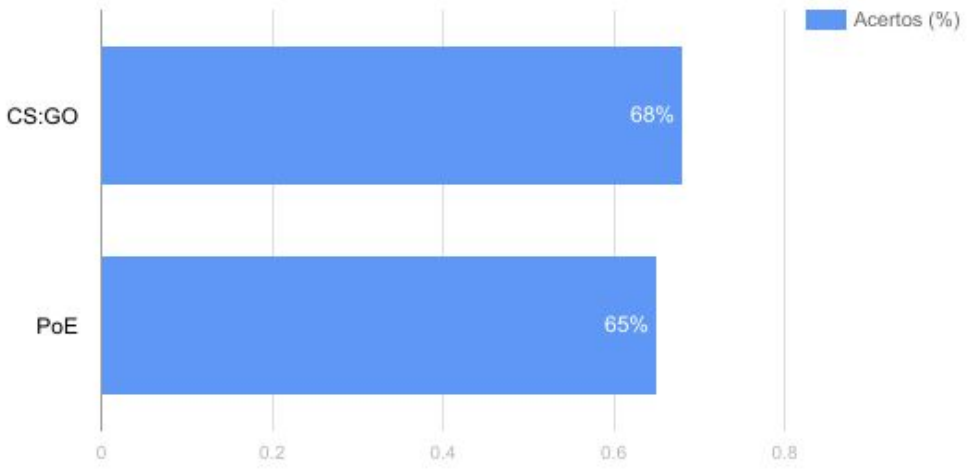

Figura 5. Porcentagem de acertos em contas classificadas com baixa confiabilidade

A implementação deste trabalho envolveu a utilização da linguagem Python [10] e das bibliotecas Graph Tool [11] e IGraph [12]. O código-fonte desenvolvido pode ser encontrado no GitHub [13].

5. Considerações Finais

A identificação de contas *Smurfs* merece uma atenção especial na elaboração de soluções. Enquanto não existem métodos determinísticos para tal, é necessário obter abordagens probabilísticas que melhor se aproximam da solução do problema. Lidar com este desafio é de extrema importância, levando em consideração que contas *Smurfs* criam situações negativas adversas em diversos tipos de redes sociais.

O presente trabalho alcançou resultados satisfatórios dado o objetivo inicial de identificar probabilisticamente usuários com contas falsas na Steam. A proposta de utilizar o SybilRank com as modificações direcionadas à rede da Steam obteve bons resultados de forma geral. Além disso, como a complexidade do algoritmo não depende da quantidade de vértices confiáveis fornecidos, e quantidade de iterações na RW são da ordem de log(n), temos um algoritmo eficiente e de alta escalabilidade.

6. Referências

[1] Cao, Q. , Sirivianos, M., Yang, X., and Pregueiro, T. (2012) “Aiding the Detection of Fake Accounts in Large Scale Social Online Services”. In Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI’12, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association

[2] “Boas-vindas ao Steam”, 2017. Disponível em: http://store.steampowered.com/. Acessado em: 6 de junho de 2017.

[3] “Facebook”, 2017. Disponível em: https://www.facebook.com/. Acessado em: 6 de junho de 2017.

[4] "Twitter. É o que está acontecendo”, 2017. Disponível em: https://twitter.com/. Acessado em: 6 de junho de 2017.

[5] Liu, L. and Webb, S. (2008) “Towards Robust Trust Establishment in Web-Based Social Networks with SocialTrust”.

[6] Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y. and Dai, Y. (2014) “Uncovering Social Network Sybils in the Wild”.

[7] “Counter-Strike: Global Offensive no Steam”, 2017. Disponível em: http://store.steampowered.com/app/730/CounterStrike_Global_Offensive/. Acessado em: 6 de junho de 2017.

[8] “Path of Exile no Steam”, 2017. Disponível em: http://store.steampowered.com/app/238960/Path_of_Exile/. Acessado em: 6 de junho de 2017.

[9] De Meo, P., Ferrarax, E., Fiumara, G., Provetti, A. (2012) “Generalized Louvain method for community detection in large networks”.

[10] “Welcome to Python.org”, 2017. Disponível em: https://www.python.org/. Acessado em: 6 de junho de 2017.

[11] “graph-tool: Efficent network analysis with python”, 2017. Disponível em: https://graph-tool.skewed.de/. Acessado em: 6 de junho de 2017.

[12] “python-igraph”, 2017. Disponível em: http://igraph.org/python/. Acessado em: 6 de junho de 2017.

[13] “Complex Network Smurfs Detection”, 2017. Disponível em: https://github.com/chriiscardozo/complexNetworkSmurfsDetection. Acessado em: 6 de junho de 2017.

Anexo A

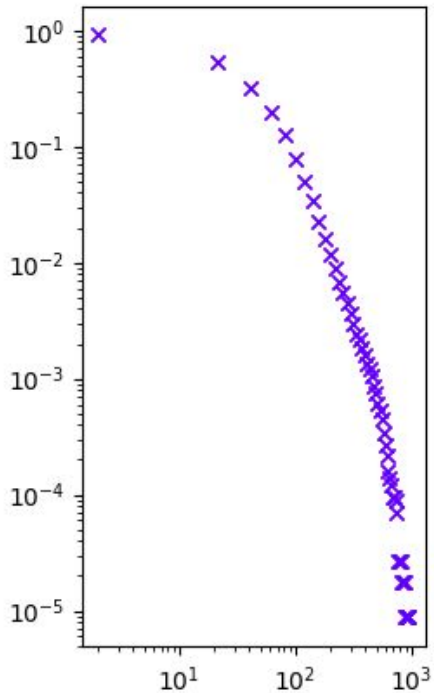

Figura A.1. CCDF empírica de grau da rede Steam.

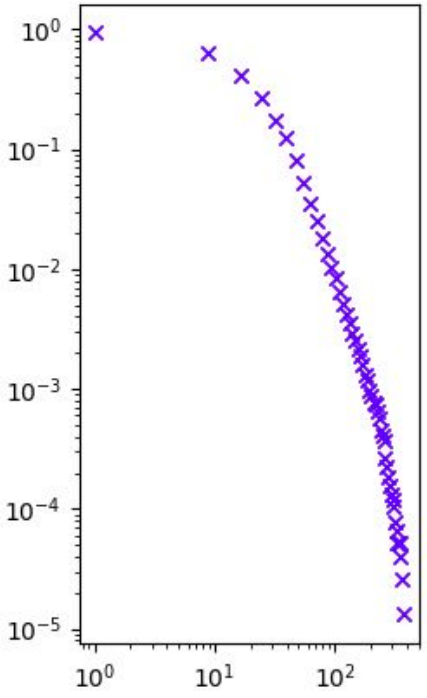

Figura A.2. CCDF empírica de grau da rede CS:GO.

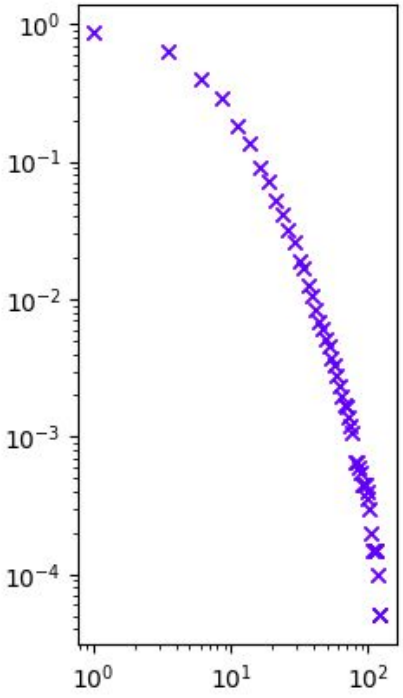

Figura A.3. CCDF empírica de grau da rede PoE.

Anexo B

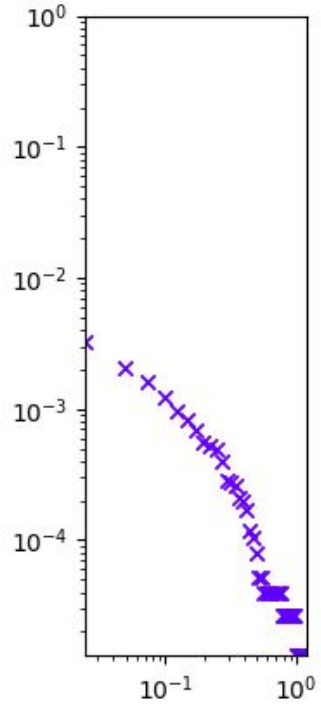

Figura B.1. CCDF empírica de ranqueamento da sub-rede CS:GO.

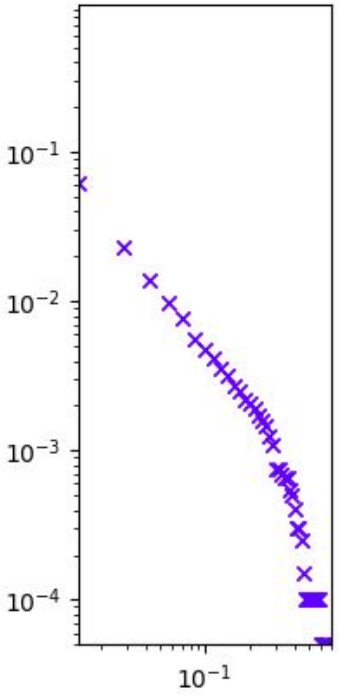

Figura B.2. CCDF empírica de ranqueamento da sub-rede PoE.