ISSN: 1983 7402 ITA, 27 a 29 SET 2022

Consciência situacional cibernética no contexto de *firmwares* de ativos de rede

Françoa Taffarel Rosário Corrêa[1], Osmany Barros de Freitas[1], Fernando Antonio Dantas Júnior[1], Gianluigi Dal Toso[1] e Lourenço Alves Pereira Júnior[1]

[1]Instituto Tecnológico de Aeronáutica — ITA
São Jose dos Campos – SP – Brazil

***Resumo*— Impulsionada pela pandemia da COVID-19, a modalidade *home office* ampliou o perímetro das redes corporativas inserindo um elo frágil na segurança: roteadores sem-fio. No entanto, essa evolução tem permitido o surgimento de uma nova ameaça cibernética às redes das infraestruturas críticas que explora vulnerabilidades nos *firmwares* dos roteadores domésticos de seus funcionários quando trabalhando remotamente. Dessa forma, o presente trabalho analisou estaticamente 45 imagens de *firmwares* de roteadores, através da ferramenta de mineração de dados DAMICORE a fim de encontrar semelhanças entre essas imagens. O resultado alcançado permitiu o agrupamento dos *firmwares* dispostos em 5 grupos que futuramente poderão ser usados a fim de potencializar ataques cibernéticos.**

***Palavras-Chave*—*firmwares*, agrupamento e vulnerabilidades.**

I. Introdução

Nos últimos anos houve um aumento nas ameaças cibernéticas às infraestruturas críticas (IC) [1], estas ameaças são obstáculos à garantia da operacionalidade das IC. Estas, tais como instalações de tratamento e distribuição de água, geração e distribuição de energia, são vitais para o bem-estar de uma sociedade. Essas instalações possuem sistemas, que são geralmente grandes, complexos e interconectados [2]. E embora tais ameaças tenham sido domínio de invasores patrocinados pelo estado, essas técnicas estão cada vez mais presentes em campanhas de *malwares* e *ransomwares* difundidas por atores não estatais.

Com o propósito de minimizar ameaças cibernéticas, nota-se que a maioria das ações de proteção de IC utilizam conceitos tradicionais de segurança de rede utilizando *firewalls* e sistemas de detecção ou prevenção de intrusão para implementar uma estratégia de defesa em profundidade. Contudo, a maioria das empresas concentra a análise de vulnerabilidades em sistemas ou softwares mais utilizados pelos seus funcionários, negligenciando as falhas de segurança intrínsecas em *firmwares* de dispositivos de redes.

Atualmente, o *firmware* de um dispositivo é um valioso recurso em se tratando de segurança cibernética. No entanto, o *firmware* muitas vezes sofre de uma ampla gama de vulnerabilidades, principalmente devido a seus recursos desatualizados ou à reutilização de bibliotecas vulneráveis. Caso comprometido nesse nível, é possível que atacantes tenham controle total das ações realizadas neste dispositivo.

Nesse contexto, devido às restrições de trabalho presencial impostas pela pandemia da COVID-19 muitas empresas adotaram o *home office* como modalidade principal. Como consequência disso, o acesso a redes corporativas foi viabilizado por meio do serviço de *Virtual Private Network* (VPN), expandindo o perímetro destas redes com a inclusão de roteadores potencialmente vulneráveis [3].

Assim, com o propósito de estudar possíveis vulnerabilidades esta pesquisa, a partir de um determinado conjunto de dados, apresenta o agrupamento *firmwares* de roteadores considerando a análise estática dos binários dos serviços mais relevantes em uma ação cibernética ofensiva.

Na sequência, este artigo está divido em oito seções, sendo a primeira esta breve introdução, a segunda relacionada à bibliografia correlata, a terceira que expõe o conjunto de dados e os métodos utilizados, a quarta que apresenta a análise descritiva dos dados, a quinta apresenta o pré-processamento de dados, a sexta expõe definição das técnicas utilizadas no agrupamento, a sétima que explica os experimentos e os resultados obtidos e por fim uma seção conclusiva sobre esta pesquisa e ações futuras relacionadas ao tema.

II. Trabalhos relacionados

Na área de segurança cibernética, o aprendizado de máquina é utilizado em estudos que visam melhorar a robustez na resposta contra incidentes cibernéticos [4]. Nesse contexto, existem várias abordagens para identificar potenciais vulnerabilidades em *firmwares* de roteadores através de técnicas tais como análise estática [5], análise dinâmica [6], execução simbólica [7]. No entanto, os trabalhos citados anteriormente não realizam o agrupamento destes *firmwares* em aspectos relacionados a segurança. Assim, nesta pesquisa, é proposto o agrupamento de imagens de *firmwares* por similaridade dos binários dos serviços mais comuns utilizados em explorações cibernéticas. Esse agrupamento permitirá, em pesquisa futura, analisar e comparar os *clusters* de *firmwares* identificados que possuam mais vulnerabilidades públicas conhecidas e, portanto, com maior superfície de ataque.

III. Materiais e Métodos

A. Conjunto de Dados

Neste artigo, a base de dados utilizada foi extraída a partir de um repositório que atualmente possui 9207 *firmwares* obtidos diretamente do site dos fabricantes, através do módulo *crawler* do *framework* ***SCREEN***- ***S****craper,* ***C****lustering,* ***RE****-hosting and* ***E****xploitatio****N*** [3]. Esta plataforma consiste em um ambiente para análise, descoberta e exploração de vulnerabilidades de *firmwares* de roteadores de maneira automática e em grande escala.

Françoa Taffarel, taffarel@ita.br; Osmany Freitas, osmany@ita.br; Fernando Dantas,fernando.dantas13@gmail.com; Gianluigi Dal Toso, gianluigi.toso@gmail.com, Lourenço Junior, ljr@ita.br.

Para o desenvolvimento desta pesquisa, foram selecionadas 45 imagens de *firmwares* e, através da utilização das ferramentas *binwalk* e *Firmadyne*, foi possível extrair valores de atributos relevantes relacionados às vulnerabilidades em roteadores. A descrição dos 10 (dez) atributos encontra-se na Tabela I, que explicita também os respectivos Tipo e Escala. Analisando esta Tabela, nota-se que foram extraídos 03 (três) atributos relacionados ao sistema operacional e 05 (cinco) relacionados a serviços de rede remotamente acessíveis, somados a outros 02 (dois) atributos de identificação: *firmware* e o *vendor* (fabricante).

TABELA I

DESCRIÇÃO DOS ATRIBUTOS COM TIPO E ESCALA.

Atributo	Descrição	Tipo	Escala
firmware	Nome do *firmware* obtido	Qualitativo	Nominal
vendor	Fabricante do roteador	Qualitativo	Nominal
busybox	Versão do pacote *busybox*	Qualitativo	Ordinal
kernel	Versão do *kernel* do *firmware*	Qualitativo	Ordinal
arch	Arquitetura do roteador	Qualitativo	Nominal
firewall	Nome e versão do *Firewall*	Qualitativo	Nominal
ssh	Nome e versão do serviço ssh	Qualitativo	Nominal
webserver	Nome e versão do servidor web	Qualitativo	Nominal
vpn	Nome e versão do servidor VPN	Qualitativo	Nominal
openssl	Versão da biblioteca OpenSSL	Qualitativo	Ordinal

B. Técnicas

Nesta pesquisa foi utilizada a técnica de agrupamento, que é uma abordagem não supervisionada dentre as atividades de aprendizado de máquina [8]. Assim, o agrupamento de dados torna-se ideal para criar uma metodologia que permita equipes potencializarem suas ações cibernéticas de exploração ou ataque, tendo em vista que o resultado final apresenta os serviços mais comuns presentes na amostra analisada.

O agrupamento tem como função encontrar estruturas em que os objetos analisados possuam similaridades relevantes no contexto de exploração dos dados [9]. O objetivo do agrupamento hierárquico é obter uma sequência aninhada de partições. Esses grupos são formados por uma matriz de similaridade, em que não há especificação prévia do número de grupos e os resultados podem necessitar de um critério de validação por especialistas [8].

Para realizar esse agrupamento hierárquico esta pesquisa fez uso da utilização da metodologia *DAta MIning of Code REpositories* (DAMICORE) [10], que combina algoritmos em outras ciências são eles: distância de compressão normalizada (NCD) da Teoria da Informação [11], Agrupamento de vizinhos (NJ) da Teoria Filogenética [12], e Algoritmo rápido de Newman (FN) da Teoria das redes complexas [13].

A ferramenta de DAMICORE tem sido utilizada com sucesso em diversas áreas, como a caracterização do perfil de consumo de recursos de programas binários [14], otimização da relação de desempenho/consumo de energia em arquiteturas *multi-cores* heterogêneas em *Field Programmable Gate Array* - FPGA [15], ou ainda para monitoramento de animais [16].

IV. ANÁLISE DESCRITIVA DOS DADOS

Nesta seção serão apresentadas as informações relacionadas à análise exploratória dos dados a fim de identificar as principais características do conjunto de dados. É valido ressaltar que esta análise é uma ferramenta muito importante, pois permite obter um resumo das principais características existentes em um conjunto de dados. O resultado obtido orienta aquilo que será realizado na futura etapa de pré-processamento.

TABELA II

AMOSTRA DA BASE DE DADOS

Atributos	**Amostras**		
firmware	F1	F30	F7
vendor	tp-link	netgear	tp-link
busybox	v1.22.1	v1.24.1	v1.22.1
kernel	3.14.77	3.14.77	3.14.43
arch	armel	armel	armel
firewall	Iptables-1.4.21	Iptables-1.4.21	Iptables-1.4.21
ssh	sshdb067	NA	sshdb
webserver	uhttpd	uhttpd/1.0.0	uhttpd
vpn	NA	OVPN2.3.2	NA
openssl	OpenSSL1.0.2d	OpenSSL1.0.2j	OpenSSL1.0.2d

A. Atributos qualitativos nominais

Os atributos qualitativos nominais considerados para a análise exploratória foram: o tipo de arquitetura (*arch*); o software e a versão do Serviço de *Secure Shell Protocol (ssh)*; o software e a versão do Serviço Virtual Private Network (*vpn*); e o software e a versão do Servidor WEB (webserver). Esses dados são muito relevantes durante a enumeração por um atacante, tendo em vista a grande quantidade de informações disponíveis sobre as vulnerabilidades das versões destes serviços.

Os gráficos de frequência dos três atributos qualitativos nominais mais relevantes para a base de dados em relação à quantidade de *firmwares* estão contidos na Fig. 1. De acordo com esses gráficos, pode-se concluir, desconsiderando os valores *not avaiable (NA)*, que a maioria dos dispositivos utiliza a arquitetura *Microprocessor without Interlocked Pipelined Stages* - MIPS.

Outro ponto a ser destacado é que a classificação de ocorrência de *firmwares* que possuem o serviço *ssh* operando o software *DropBear*, que é projetado para sistemas com poucos recursos computacionais, sendo uma alternativa ao *OpenSSH*, padrão em sistemas operacionais UNIX. Soma-se ainda que a maioria dos *firmwares* utiliza o serviço *uhttpd* como principal servidor web, refletindo que, caso exista uma vulnerabilidade, a probabilidade de ataque em larga escala aumenta.

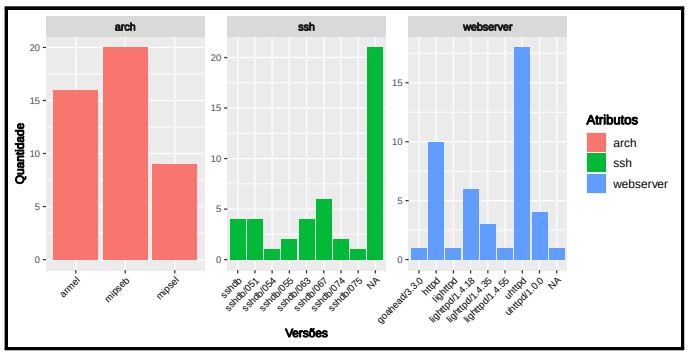
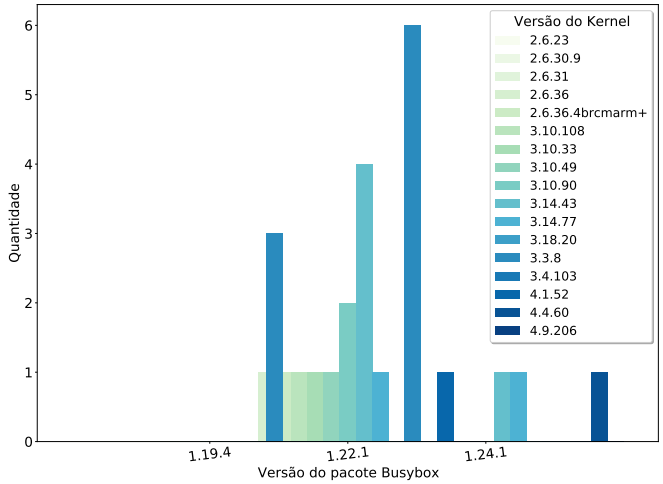


Fig. 1. Frequência de três atributos qualitativos nominais da base de dados.

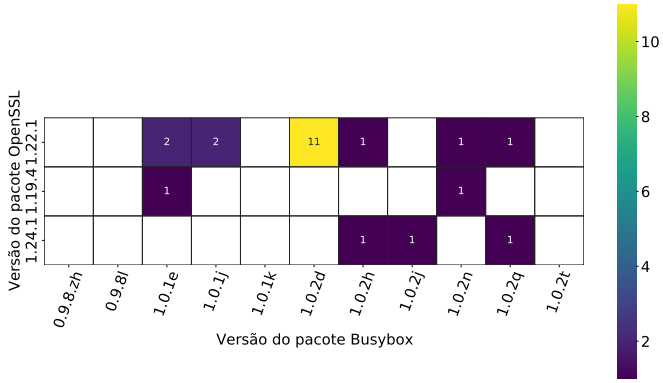
B. Atributos qualitativos ordinais

Na Fig. 2a estão representados os gráficos com análises multivariadas entre os 03 (três) atributos qualitativos ordinais:

versão do *Kernel* e versão do pacote *BusyBox* e versão do *OpenSSL*. Na Fig. 2b, à direita, é apresentado um *heatmap* que relaciona versões do binário *BusyBox* [17] com binário *OpenSSL*, uma biblioteca que implementa as funções básicas de criptografia. É possível notar que a maioria dos *firmwares* que possui *OpenSSL* apresentam também a versão *v.1.22.1* da *BusyBox*. Por fim, analisando o gráfico à esquerda, desconsiderando os dados ausentes relativos ao atributo *Kernel*, nota-se que, na maioria dos *firmwares* que possui OpenSSL, predomina a versão acima da 3.0.



a) Kernel e BusyBox



b) BusyBox e OpenSSL

Fig. 2. Gráficos relativos aos três atributos qualitativos ordinais.

V. Pré-processamento dos dados

Essa seção abordará a etapa de pré-processamento dos dados que é composta por um conjunto de atividades que envolvem realizar a preparação, a organização e a estruturação dos dados. Essas ações têm o propósito de tornar os dados mais apropriados para a utilização na combinação dos algoritmos citados na subseção 3.2.

Assim, nesta etapa foram utilizadas algumas técnicas citadas por [18], dentre as quais foram utilizadas a transformação de dados e eliminação manual do atributo *vendor*, tendo em vista as características a serem exploradas em uma ação de guerra cibernética independem do tipo de *vendor*. Ressalta-se que, a técnica de eliminação de valores ausentes, comumente utilizada em artigos relacionados a aprendizado de máquina, não foi utilizada neste trabalho em virtude da relevância no agrupamento por similaridades, pois o fato do *firmware* não possuir um determinado serviço é significante para a decisão de uma ação de exploração cibernética. Em seguida, é exposto a descrição dos passos que foram utilizados durante a etapa de pré-processamento dos dados.

A. Transformação dos atributos qualitativos nominais

Em sua totalidade a base de dados apresenta atributos qualitativos e a maioria destes atributos são nominais, os quais evidenciam diferentes versões e softwares relacionados aos serviços presentes nos *firmwares*. Para este estudo, foi definido que todos os textos da base seriam transformados em minúsculo. Dessa forma, referente aos atributos nominais foi realizada a transformação de dados de softwares iguais através da padronização de nomes para o formato *nome do software/nº da versão*, exemplo *Iptables-1.4.21* para *iptables/1.4.21*, esse padrão foi adotado para todos os demais atributos nominais. Essa padronização de capitalização é necessária para evitar inconsistências com algoritmos de agrupamento que possuem sensibilidade a letras com diferenças na capitalização [19]. Vale ressaltar que softwares encontrados nos *firmwares*, mas sem constatação de suas versões, foram listados somente pelo seu nome na base de dados, exemplo *uhttpd*.

B. Transformação dos atributos qualitativos ordinais

Os atributos não transformados anteriormente são qualitativos ordinais, pois tratam-se de um mesmo software com versões diferentes. De forma igual, os valores desses atributos foram transformadas para minúsculo e foi realizada a transformação de dados para o formato *nome ou nº da versão* sem qualquer outra letra anterior ao primeiro número, por exemplo em relação aos atributos *busybox* e *openssl*, *v1.22.1* para *1.22.1* e *OpenSSL1.0.2d* para *1.0.2d*, respectivamente. Esse padrão foi adotado para todos os demais atributos.

TABELA III
AMOSTRA DO CONJUNTO DE DADOS PRÉ-PROCESSADO

Atributos	**Amostras**		
firmware	F1	F30	F7
busybox	1.22.1	1.24.1	1.22.1
kernel	3.14.77	3.14.77	3.14.43
arch	armel	armel	armel
firewall	iptables/1.4.21	iptables/1.4.21	iptables/1.4.21
ssh	sshdb/067	NA	sshdb
webserver	uhttpd	uhttpd/1.0.0	uhttpd
vpn	NA	ovpn/2.3.2	NA
openssl	1.0.2d	1.0.2j	1.0.2d

Nesse contexto, a Tabela III apresenta uma amostragem do conjunto de dados pré-processado. Após a conclusão desta etapa, o conjunto de dados foi exportado para um arquivo do tipo CSV, denominado *firmware_dataset_ppd.csv*. Os dados deste arquivo foram processados pelo DAMICORE, que será apresentado na próxima seção, contudo para correta utilização desta ferramenta foi necessário criar 45 novos arquivos localizados em [20].

VI. Definição das tarefas de aprendizado de agrupamento

As técnicas fundamentais usadas nestas pesquisa foram escolhidas a fim de permitir a aplicação do DAMICORE. Esta ferramenta produz agrupamentos expostos através de relações

hierárquicas, em quaisquer tipos de dados, sem a necessidade de um estudo sematológico entre eles. As próximas seções descrevem cada um dos métodos que são compilados dentro da ferramenta.

A. Distância de Compressão Normalizada (NCD)

A Distância de Compressão Normalizada [11] pode ser utilizada para diversos tipos de dados. A ideia é que: dados dois objetos *a* e *b*, os mesmos são considerados próximos se *a* pode ser significativamente compactado usando a informação em *b*, e vice-versa, ou seja, se duas entidades são semelhantes é mais fácil para descrever a segunda apenas referindo suas diferenças em relação à primeira.

As propriedades de tal compressão são a base da teoria da complexidade de Kolmogorov [21]. No entanto, o cálculo da complexidade Kolmogorov é computacionalmente intratável. Felizmente, o NCD foi proposto como uma métrica baseada em compressão que aproxima a complexidade de Kolmogorov em um tempo computacional razoável [11], sendo definida a seguir (1):

$$NCD(x;y) = \frac{C(xy) - min\{C(x);C(y)\}}{max\{C(x);C(y)\}} \quad (1)$$

onde *C(x)* é o comprimento da versão compactada do arquivo *x* obtido ao utilizar um compressor pré-definido, e *C(xy)* é o arquivo resultante da concatenação de *x* e *y*.

B. Agrupamento de vizinhos (NJ)

A partir dos valores de distância dos dados de entrada, o algoritmo *Neighbor-Joining* (NJ) será capaz de identificar semelhanças hierárquicas e produzir uma árvore filogenética [12]. O NJ apresenta uma ótima relação custo/benefício para lidar com uma diversidade relativamente grande de tipos de dados [10].

C. Algoritmo rápido de Newman (FN)

A saída do NJ é, portanto, uma árvore filogenética de relacionamentos, sendo necessário outro método para extrair potenciais *clusters* escondidos na topologia da árvore. Para realizar esta ação será empregado o algoritmo rápido de Newman (FN). É um algoritmo que possui um alto grau de escalabilidade para a manipulação de um conjunto de dados adequadamente dispostos em um modelo de rede, permitindo agrupá-los por similaridade [13].

VII. Experimentos e Resultados

Dessa forma, como base para o experimento será utilizada o DAMICORE em sua versão na linguagem *python* [22]. A ferramenta analisará o conjunto de dados pré-processado citado na subseção 5.2 através de três passos utilizando, respectivamente, os algoritmos citados nas subseções anteriores: o primeiro passo consiste na construção de uma matriz de distâncias formada pela comparação par a par de cada um dos objetos do conjunto levando em consideração uma métrica de similaridade por Distância de Compressão Normalizada; o segundo passo consiste na conversão dessa matriz de distâncias em uma rede de similaridades através do algoritmo NJ exposta em forma de árvore filogenética; por fim uma vez encontrada a filogenia dos dados, será então verificada a existência de comunidades com alto grau de similaridade nesta árvore, através da aplicação do FN a fim de evidenciar grupos com elementos semelhantes distribuídos de forma hierárquica.

A. Avaliação dos resultados do DAMICORE

O DAMICORE propôs de forma hierárquica um particionamento inicial dos dados em grupos. A Fig. 3 contém o gráfico com a distribuição dos *firmwares* nos agrupamentos encontrados pelo DAMICORE. Foram detectados 16 grupos, com mais de um elemento, sendo eles: A, B, C, D, F, G, H, I, J, K, L, M, N, O, P e Q. Nota-se nos grupos G, H, I, M e P que há semelhança entre os *firmwares* de diferentes fabricantes do repositório. Por outro lado, os grupos A, B, C, D, E, F, J, K, L, N, O e Q possuem amostras apenas de 01 (um) fabricante.

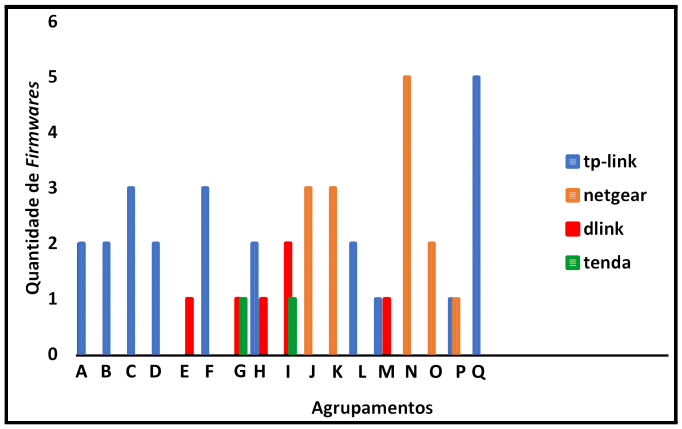


Fig. 3. Distribuição dos *firmwares* nos agrupamentos por fabricante.

A Fig. 4 contém a árvore filogenética gerada para o conjunto de dados. Os ramos estão coloridos de acordo com o nome dos fabricantes. É notório que há ramos com amostras exclusivas de fabricantes como Netgear (laranja) e TP-Link (azul). Enquanto que *firmwares* do fabricante Tenda (verde) e D-LINK (vermelho) ficaram distribuídos em outros ramos da árvore, sem possuir, necessariamente, um agrupamento exclusivo. Esses ramos mais uniformes, onde há predominância de um determinado fabricante, evidenciam que, apesar de modelos e versões de *firmware* diferentes, há uma repetição no padrão adotado pelas empresas na escolha dos serviços implementados por seus produtos.

Determinar o número ótimo de grupos tem sido um grande desafio para os especialistas, tendo em vista que este valor depende da abordagem usada para medir as dissimilaridades/semelhanças e as variáveis usadas para agrupamentos [23]. Uma resposta despretensiosa consiste em examinar o dendrograma moldado com o auxílio de agrupamento hierárquico para perceber se ele apresenta um número específico de agrupamentos.

Contudo, antes de realizar essa ação e com o propósito de encontrar um número ideal de grupos, esta pesquisa realizou a aplicação dos seguintes critérios de validação presentes no pacote *Factoextra* utilizado para identificar o número ideal de *clusters* através da função *fviz_nbclust* da linguagem R: critério Cotovelo (*Elbow method*) [24], critério Silhueta (*Silhouettes method*) [25] e Estatística GAP (*Gap Static method*)[26].

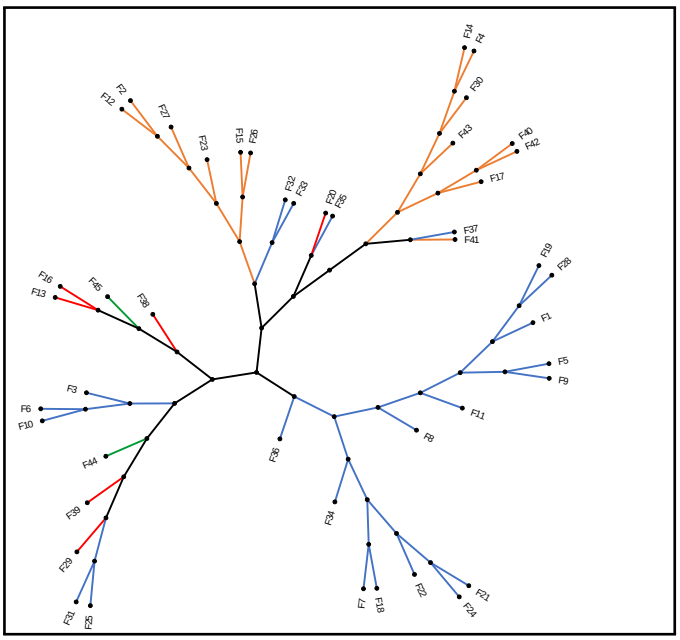


Fig. 4. Árvore filogenética do repositório.

Nesse contexto, ambos os critérios necessitam de um valor *k* referente ao número máximo de *clusters* esperados para estimar o número ótimo de agrupamentos. Para o valor de máximo *k=5*, foi possível estimar, o número ideal de *clusters* foram [4, 3 e 4], respectivamente, para cada um dos critérios citados na Fig. 6. O dendrograma é formado por camadas de nós, na qual cada uma é a representação de um grupo [8]. Embora os critérios de validação aplicados tenham resultado em valores entre 3 e 4 como número ótimo de agrupamentos, a partir da análise visual do dendrograma gerado pelo DAMICORE, exposto na Fig. 5, é possível verificar que a otimização dos *clusters* pode ser obtida com um corte no nível $k = 5$, produzindo um total de cinco grupos, conforme apresentados na Tabela IV.

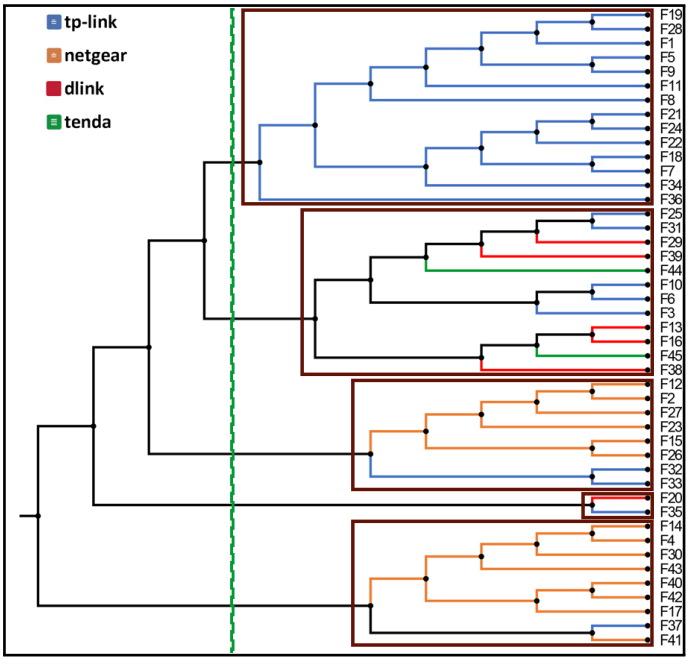
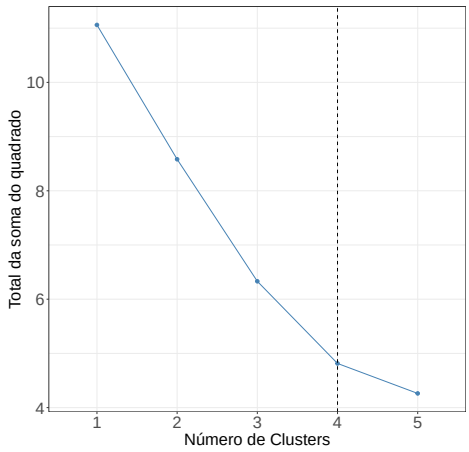
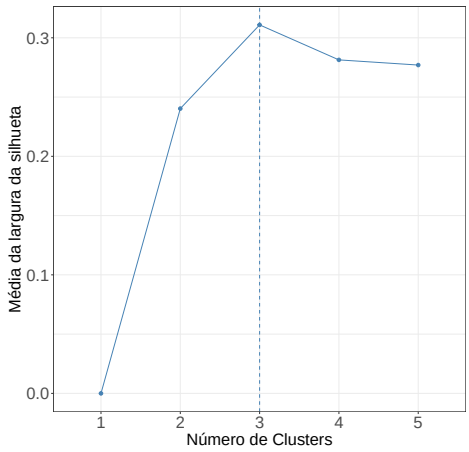


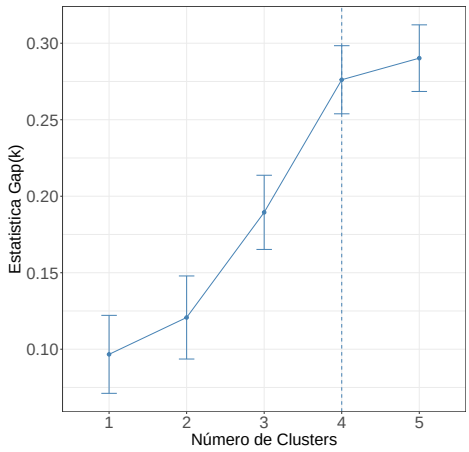
Fig. 5. Dendrograma gerado para o repositório.



a) Elbow method



b) Silhouette method



c) Gap statistics method

Fig. 6. Estimação do número ótimo de *cluster* através dos métodos através do pacote Factoextra.

A Tabela IV reflete o resultado observado na árvore filogenética, onde nota-se o *Cluster* 1 com agregação exclusiva do fabricante TP-Link e os *Clusters* 3 e 5 com expressiva predominância da Netgear. Por outro lado, no *Cluster* 2 bastante heterogêneo, em se tratando de fabricantes, num total de 03, porém observa-se que os serviços implementados por estas amostras se assemelham em suas versões, tornando válida o agrupamento atribuído. Por fim, nota-se que no *Cluster* 4 há exclusividade de 2 amostras de *firmwares*, nas quais foi observado que as versões dos

serviços implementados divergem dos demais existentes no repositório, porém se assemelham entre si em mais de 70%, sendo considerado um resultado exitoso pela ferramenta.

TABELA IV
ANÁLISE VISUAL DO DENDROGRAMA COM UM CORTE EM $k = 5$.

Grupos	Firmwares			
Cluster 1	F19	F28	F1	F5
	F9	F11	F8	F21
	F24	F22	F18	F7
		F34	F36	
Cluster 2	F12	F2	F27	F23
	F15	F26	F32	F33
Cluster 3	F25	F31	F29	F39
	F44	F10	F6	F3
	F13	F16	F45	F3
Cluster 4		F20	F35	
Cluster 5	F14	F4	F30	F43
	F42	F17	F37	F41

VIII. CONCLUSÃO

Este trabalho analisou os dados extraídos de um subconjunto das imagens de *firmwares* disponíveis no repositório do *framework* SCREEN. O objetivo foi a utilização de técnicas de agrupamento para avaliar a capacidade de identificação de grupos de similaridade entre as imagens analisadas. Tal procedimento foi executado com êxito através da execução da ferramenta DAMICORE, que reúne os algoritmos NCD, FN e NJ.

Por se tratar de uma tarefa não supervisionada, não existe uma única estrutura de agrupamentos dada como correta, sendo possível outras representações do mesmo conjunto de dados. Neste trabalho, o resultado do processamento do DAMICORE resultou em 17 grupos contendo de 01 a 05 *firmwares*, conforme Fig. 3.

Em seguida, foi realizada outra análise no quinto nível do dendrograma. Esta ação ocorreu com o propósito de maximizar a obtenção de grupos válidos tomando por base os critérios de estimativa do número ótimo de *cluster* e a baixa cardinalidade apresentada como resultado do DAMICORE. Deste resultado, foi possível identificar cinco novos grupos com forte semelhança nas versões dos serviços adotados por cada empresa. Como consequência, a descoberta de vulnerabilidade em algum destes serviços, sugere o comprometimento de outros *firmwares* do mesmo grupo.

Como trabalhos futuros pretende-se identificar serviços com versões vulneráveis e utilizá-los como critério de agrupamento, objetivando agrupar *firmwares* pela existência de falhas passíveis de exploração por um atacante. Tal informação será de fundamental relevância para o módulo de *Clustering* do *framework* SCREEN, que será responsável por alimentar outro módulo chamado *Exploitation* [3].

REFERÊNCIAS

[1] N. Chowdhury and V. Gkioulos, "Cyber security training for critical infrastructure protection: A literature review," *Computer Science Review*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013721000010

[2] W. Gao, T. Morris, B. Reaves, and D. Richey, "On scada control system command and response injection and intrusion detection," in *2010 eCrime Researchers Summit*, 2010.

[3] G. D. Toso and L. A. Pereira, "Enumeration of operating systems and services in the wireless router's firmware context." in *Anais Estendidos do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2021)*. Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2021.

[4] M. Mirzaie and M. Nooraei Abadeh, "An approach to analyze the vulnerability of function-based social networks using clustering coefficient," *Tabriz Journal of Electrical Engineering*, 2020. [Online]. Available: https://tjee.tabrizu.ac.ir/article_10750.html

[5] Q. Feng, R. Zhou, C. Xu, Y. Cheng, B. Testa, and H. Yin, "Scalable graph-based bug search for firmware images," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2976749.2978370

[6] D. D. Chen, M. Woo, D. Brumley, and M. Egele, "Towards automated dynamic analysis for linux-based embedded firmware," in *NDSS*, 2016.

[7] D. Babić, L. Martignoni, S. McCamant, and D. Song, "Statically-directed dynamic automated test generation," in *Proceedings of the 2011 International Symposium on Software Testing and Analysis*, ser. ISSTA '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/2001420.2001423

[8] K. Faceli, A. C. Lorena, J. Gama, T. A. Carvalho, Almeida, and A. C. P. de Leon Ferreira, *Inteligência artificial: uma abordagem de aprendizado de máquina*, 2nd ed. LTC, 2021.

[9] R. Dubes and A. Jain, "Clustering methodologies in exploratory data analysis," in *Advances in Computers*, ser. Advances in Computers, M. C. Yovits, Ed. Elsevier, 1980. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0065245808600340

[10] A. Sanches, J. M. Cardoso, and A. C. Delbem, "Identifying merge-beneficial software kernels for hardware implementation," in *2011 International Conference on Reconfigurable Computing and FPGAs*, 2011.

[11] R. Cilibrasi and P. Vitanyi, "Clustering by compression," *IEEE Transactions on Information Theory*, 2005.

[12] J. Felsenstein, *Inferring Phylogenies*, 2nd ed. Sunderland, Mass: Sinauer Associates, Sep. 2003.

[13] M. Newman, *Networks: An Introduction*. USA: Oxford University Press, Inc., 2010.

[14] R. Pinto, A. Delbem, and F. Monaco, "Caracterização do perfil de consumo de recursos de programas binários utilizando a técnica damicore," in *Anais do XIII Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS, Brasil: SBC, 2017. [Online]. Available: https://sol.sbc.org.br/index.php/sbsi/article/view/6034

[15] B. d. A. SILVA, "Um método de otimização da relação desempenho/consumo de energia para arquiteturas multi-cores heterogêneas em fpga," Tese de Doutorado em Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.

[16] J. F. C. Camargo, "Desenvolvimento de tecnologia de hardware e software para o monitoramento de animais," Dissertacao de Mestrado em Matemática, Estatística e Computação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2019.

[17] N. Wells, "Busybox: A swiss army knife for linux," *Linux J.*, vol. 2000, 2000.

[18] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining (2nd Edition)*, 2nd ed. Pearson, 2018.

[19] L. N. d. Castro and D. G. Ferrari, *Introdução à mineração de dados: conceitos básicos, algoritmos e aplicações*, 1st ed. Saraiva Uni, 2016.

[20] x, "Clustering firmwares," https://github.com/c2dc/screen-clustering, 2022.

[21] M. Li and P. M. Vitnyi, *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed. Springer Publishing Company, Incorporated, 2008.

[22] B. K. M. CESAR, "Estudo e extensão da metodologia damicore para tarefas de classificação," Dissertação de Mestrado em Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo,São Carlos, 2016.

[23] S. Zhou, Z. Xu, and F. Liu, "Method for determining the optimal number of clusters based on agglomerative hierarchical clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[24] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, 1953. [Online]. Available: https://doi.org/10.1007/BF02289263

[25] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, 1987.

[26] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2001.