Identificação Remota de Dispositivos IoT via Análise da Dinâmica Temporal de Números de Sequência do TCP

Manoel Anízio Azevedo de Oliveira[1], Luiz Paulo de Assis Barbosa[1], Antonio Alfredo Ferreira Loureiro[2], João Paulo de Souza Medeiros[1], João Batista Borges[1]

[1]Departamento de Computação e Tecnologia (DCT)
Universidade Federal do Rio Grande do Norte (UFRN) – Caicó – RN – Brasil

[2]Departamento de Ciência Computação (DCC)
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

manoel.oliveira.066@ufrn.edu.br, loureiro@dcc.ufmg.br

{luiz.paulo, joao.paulo.medeiros, joao.borges}@ufrn.br

Abstract. *Due to their computational restrictions and incorrect configurations, devices in the Internet of Things (IoT) are easy targets for various attacks. In this paper, a novel approach based on the ordinal patterns transformations is proposed for the remote identification of Operating Systems (OSs), a fundamental step to identify possible vulnerabilities in these devices. For that, the dynamic behavior of the initial sequence numbers (ISN), contained within the header of the Transmission Control Protocol (TCP), is analyzed. We verified the method's ability to detect similarities and differences between classic and modern OSs, comparing them with IoT devices. The experiments prove its effectiveness in recognizing OSs by their different ISN generation patterns, outperforming consolidated tools such as Nmap, as well as being able to classify them with 100% accuracy in certain cases.*

Resumo. *Devido às suas restrições computacionais e configurações incorretas, os dispositivos da Internet das Coisas (IoT) são alvos fáceis de diversos ataques. Neste trabalho, propomos uma nova abordagem baseada nas transformações de padrões ordinais para a identificação remota de Sistemas Operacionais (SOs), uma etapa fundamental para identificar possíveis vulnerabilidades nesses dispositivos. Para isso, é analisado o comportamento dinâmico dos números iniciais de sequência (ISN) do cabeçalho do TCP. Verificamos a capacidade do método na detecção de similaridades e diferenças entre SOs clássicos e modernos, comparando com dispositivos da IoT. Os experimentos comprovam sua eficácia em reconhecer os SOs pelos seus diferentes padrões de geração de ISN, superando ferramentas consolidadas como o Nmap, bem como sendo capaz de classificá-los com uma acurácia de 100% em certos casos.*

1. Introdução

O processo de identificação remota de Sistemas Operacionais (SOs), do inglês, *OS fingerprinting*, se dá por meio da análise de pacotes originados de um sistema alvo, em uma rede de computadores, com o intuito de extrair alguma característica de seu

SO [Medeiros et al. 2010]. Essa identificação é importante pois a versão do sistema operacional e de seus serviços pode oferecer uma visão mais precisa sobre possíveis vulnerabilidades [Beck et al. 2007]. A principal ferramenta para a identificação remota de SOs é o Nmap, lançada em 1997 por Gordon Lyon [Lyon 2009]. O Nmap utiliza uma série de características extraídas de vários SOs, que são guardadas em uma base de dados de assinaturas, para comparar e fazer a sua distinção. No entanto, apesar de ser a forma mais utilizada por profissionais de segurança, essas bases de dados que, em grande parte, são construídas pela comunidade usuária da ferramenta, encontram-se desatualizadas. Além disso, a quantidade de amostras de assinaturas novas tem caído nos últimos anos.

A quantidade de características coletadas pelo Nmap no momento da construção da assinatura precisa ser suficientemente grande e diversa para identificar unicamente um SO. Assim, o Nmap envia uma série de pacotes para a máquina alvo para obter tais informações. Além disso, por coletar informações das camadas de rede e transporte, o Nmap pode ser afetado por *firewalls* ou roteadores, podendo interferir nos dados coletados. Há outras iniciativas que buscam reduzir a quantidade de pacotes enviados para a máquina alvo. Por exemplo, é possível enviar apenas um único pacote [Shamsi et al. 2016], e utilizar os tempos subsequentes de retransmissões para identificar o seu SO. Porém, variações nos tempos de resposta devido às condições da rede (e.g., *jitter*) e perdas de pacotes são problemas aos quais esses métodos estão sujeitos.

Outra estratégia para identificação remota de SOs utiliza apenas uma informação do cabeçalho *Transmission Control Protocol* (TCP), o seu número de sequência inicial, do inglês, *Initial Sequence Number* (ISN). O número de sequência é um campo que contém informações utilizadas para a identificação dos dados transmitidos pelo TCP. No entanto, conforme definido na RFC 9293 [Eddy 2022], em seu primeiro pacote de sincronização, essa informação consiste no ISN, que é utilizado como um identificador inicial para a ordem dos demais pacotes na conexão. Portanto, conforme proposta de [Medeiros et al. 2010], são enviados pacotes TCP SYN para o sistema alvo, em intervalos constantes de tempo, com o intuito de estabelecer novas conexões, sendo coletados os valores de ISN respondidos. Para cada nova conexão, um novo valor de ISN é gerado pelo sistema que, para evitar vulnerabilidades de segurança, utiliza alguma forma de geração por meio de números pseudo-aleatórios [Gont and Bellovin 2012].

Essa forma de geração insere nos valores do ISN um componente de aleatoriedade que, possivelmente, influenciará a sua dinâmica temporal, ou seja, a forma como esses valores evoluem no tempo [Rosso et al. 2007, Borges et al. 2022]. Assim, neste trabalho, propomos uma nova abordagem que tem como foco a análise do comportamento dinâmico dos valores de ISN presentes no cabeçalho TCP, por meio da utilização das transformações de padrões ordinais [Rosso et al. 2007]. Os padrões ordinais, propostos por [Bandt and Pompe 2002], consistem na transformação de uma série temporal em símbolos, obtidos analisando a relação ordinal entre valores sucessivos. Neste trabalho, propomos a utilização das transformações de padrões ordinais para a identificação remota de SOs, que, até onde sabemos, ainda não foi aplicada neste cenário.

Como mencionado acima, o problema investigado neste trabalho é relevante pois uma das informações mais importantes em um ataque é saber o ambiente computacional de um dispositivo IoT. Além disso, com a evolução dos SOs e a criação de novas versões e implementações de suas pilhas de protocolos de comunicação, bem como a grande quan-

tidade e variedade de sistemas para os dispositivos da Internet das Coisas, as ferramentas atuais não são mais capazes de identificar corretamente tais sistemas. Adicionalmente, devido às suas restrições computacionais e configurações incorretas, os dispositivos da IoT são alvos fáceis de diversos ataques [Bertino and Islam 2017]. Portanto, dada a inserção e importância destes dispositivos em nossas atividades do cotidiano, torna-se urgente a proposição de soluções para sua segurança [Borges et al. 2023].

Desta forma, com o intuito de dar novos direcionamentos para as estratégias de identificação de dispositivos, servindo como uma camada a mais de informações, as principais contribuições deste trabalho são (i) apresentar uma nova abordagem para identificação de SOs por meio da análise de ISNs utilizando a transformação de padrões ordinais, e (ii) analisar sua eficácia na identificação e classificação de diferentes SOs, considerando versões clássicas e modernas, em comparação com dispositivos IoT.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta trabalhos relacionados e a Seção 3 detalha nossa fundamentação teórica. A Seção 4 apresenta a proposta deste trabalho. A Seção 5 detalha os experimentos e a análise dos resultados obtidos. Por fim, a Seção 6 conclui o trabalho e discute possíveis extensões.

2. Trabalhos Relacionados

A principal ferramenta utilizada pela comunidade para a identificação remota de SOs é o Nmap [Lyon 2009]. Além do Nmap, há outras ferramentas para *OS Fingerprinting*, como p0f, Ettercap, NetSleuth, NetworkMiner e Satori. Entretanto, com exceção da Satori, as demais possuem bases de dados defasadas, com a atualização mais recente e significativa sendo do p0f em 2014 [Laštovička et al. 2023]. Outras estratégias vão na direção oposta ao Nmap, buscando reduzir a quantidade de características extraídas, como o trabalho de [Shamsi et al. 2016]. Os autores propõem um método de identificação de SOs utilizando apenas um pacote SYN. Contudo, apesar de conseguirem uma acurácia de 98%, seu método pode ser seriamente impactado por problemas como a perda de pacotes e alterações nos tempos de envio e recebimento do pacote. O trabalho proposto por [Medeiros et al. 2010] utiliza apenas uma única informação da máquina alvo, seus valores de ISN, extraídos de uma única porta TCP aberta. Nesse trabalho, os autores estudam a geração de ISNs por diferentes SOs, a fim de utilizá-los na criação de assinaturas para realizar o *OS Fingerprinting*, demonstrando resultados positivos com a sua classificação associada a algoritmos de *Machine Learning* (ML).

Por sua vez, [Ordorica 2017] utiliza uma abordagem passiva para a identificação de SOs, utilizando características extraídas do tráfego IPv6 e classificando os SOs por meio do algoritmo de *Random Forest* (RF). [Song et al. 2019] fazem um estudo de quatro métodos de identificação de SOs: um baseado em regras, um algoritmo de Árvore de Decisão, um algoritmo de *K-nearest neighbors* (KNN) e outro de redes neurais artificiais, sendo esse último escolhido para ser comparado com a ferramenta *NetworkMiner*. [Fan et al. 2022] propõem um retorno à utilização de diversas métricas, similar ao Nmap, para a identificação de SOs, baseando-se na utilização de um algoritmo RF hierárquico.

Com relação aos estudos de padrões ordinais, os encontramos aplicados em diversas áreas do conhecimento, desde finanças até aplicações meteorológicas [Borges et al. 2022]. No entanto, para a identificação remota de SOs, até onde sabemos, esse método ainda não foi aplicado. Para sua aplicação, é necessária uma série temporal

de amostras, conforme definido por [Bandt and Pompe 2002]. Assim, dentre os trabalhos citados acima, seguiremos a estratégia do trabalho proposto por [Medeiros et al. 2010]. Nesse trabalho, os autores associam a dinâmica caótica da sequência de ISNs a atratores de espaço de fase, sendo analisado o comportamento temporal desses valores. Desta forma, essa abordagem atende aos requisitos da transformação de padrões ordinais, gerando como assinatura dos SOs uma série temporal com consecutivos valores de ISN. Para avaliar o potencial do método, comparamos com os resultados da ferramenta Nmap, por ser a mais popular e abranger mais SOs do que as outras ferramentas citadas.

3. Fundamentação Teórica

Neste trabalho, propomos a utilização da transformação de padrões ordinais para a identificação remota de SOs, por meio da análise do comportamento dinâmico de seus valores de ISN. Nesta seção, é apresentada uma visão geral sobre esses fundamentos.

3.1. Sequence Number (SN)

O número de sequência consiste em identificadores únicos de 32 bits atribuídos ao primeiro segmento de dados após as informações de porta de destino e origem em uma conexão TCP. Todo pacote enviado em uma conexão TCP contém um SN. Esse identificador é gerado durante o início de uma conexão TCP via o *Three-way Handshake*. O primeiro SN gerado é chamado de *Initial Sequence Number* (ISN) e deve ser gerado utilizando elementos de aleatoriedade. Um ISN não-aleatório deixa brechas para inúmeras falhas de segurança [Gont and Bellovin 2012]. Se alguém mal intencionado consegue prever o próximo ISN em uma comunicação TCP, pode-se interceptar e tomar controle da sessão.

Conforme [Eddy 2022], na RFC 9293, os ISNs devem ser gerados usando uma sequência de números chamada de "*clock*", um contador de 32 bits que é incrementado a cada 4ms monotonicamente, ou seja, o próximo é sempre maior que o anterior. Além do *clock*, a RFC recomenda que os ISNs sejam gerados utilizando uma expressão aritmética:

$$N_{\text{ISN}}(t) = M(t) + F(\textit{localip}, \textit{localport}, \textit{remoteip}, \textit{remoteport}, \textit{secretkey}), \qquad (1)$$

onde $M(\cdot)$ é o *clock*, $F(\cdot)$ é uma função pseudoaleatória, *localip, localport, remoteip, remoteport* são os identificadores da conexão e *secretkey* são algum tipo de dados criptografados. A maneira como a *secretkey* deve ser encriptada não é descrita pela RFC. Sendo assim, os Sistemas Operacionais podem utilizar de diferentes algoritmos de encriptação, e como consequência, diferentes maneiras de gerar o ISN.

3.2. Padrões Ordinais

A transformação em padrões ordinais, definida originalmente por [Bandt and Pompe 2002], são utilizados para analisar séries temporais, onde valores sucessivos são analisados e transformados em símbolos. A frequência desses padrões é então usada para caracterizar as dinâmicas dessa série [Zanin 2023]. Esse método se baseia em dois parâmetros: a dimensão D e o intervalo τ. Esses parâmetros mapeiam os valores da série em $D!$ possíveis símbolos, em janelas de intervalo τ e tamanho D.

Portanto, seja $X = \{x_1, ..., x_n\}$ uma série temporal de tamanho T, é possível descrever a transformação de Bandt-Pompe como um procedimento de duas etapas: particionamento e permutação [Chagas et al. 2022]. Para o particionamento, é criada uma

janela w de dimensão D e intervalo τ, com a forma

$$w_t^{(D,\tau)} = (x_t, x_{t+\tau}, ..., x_{t+(D-1)\tau}), \tag{2}$$

para $t = 1, 2, ..., N$, onde $N = T - (D-1)\tau$.

Na permutação, os padrões são obtidos avaliando os índices de cada partição. Cada padrão é gerado através do índice que ordenaria cada elemento da partição (ordem crescente). Por exemplo, para a partição $(9, 10, 6)$, o padrão ordinal gerado seria $(3, 1, 2)$.

Após gerar todos os padrões ordinais da série, podemos atribuir a distribuição de probabilidade p_π para cada uma das permutações. Então, seja o conjunto Π de padrões ordinais, e π_t cada uma das possíveis permutações desses padrões, onde $t = 1, 2, ..., D!$, e $|s_{\pi_t}|$ o número de padrões do tipo π_t, sua distribuição de probabilidade é dada por

$$p(\pi_t) = \frac{|s_{\pi_t}|}{n - (D-1)\tau}. \tag{3}$$

3.3. Métricas de Teoria da Informação

A partir da distribuição de probabilidade p_π, faz-se possível computar métricas de teoria da informação, capazes de representar diferentes características da série temporal que a originou. A seguir, apresentamos as métricas que utilizamos neste trabalho.

Entropia de Permutação de Shannon A Entropia de Permutação de Shannon mede o grau de incerteza de um sistema. Pode ser calculada com os valores $p(\pi)$ obtidos da distribuição p_π, definindo-se como:

$$H[p_\pi] = -\sum p(\pi) \log p(\pi). \tag{4}$$

Entretanto, é comum optar-se por sua forma normalizada [Chagas et al. 2022]:

$$H_s[p_\pi] = \frac{H[p_\pi]}{H_{\text{max}}}, \tag{5}$$

onde $H_{\text{max}} = H[p_u] = \log D!$, e $p_u = \{1/D!, ..., 1/D!\}$.

Complexidade Estatística A complexidade estatística desconta a aleatoriedade de um sistema e fornece uma medida para a regularidade presente nele [Feldman and Crutchfield 1998]. É baseada na divergência de Jensen-Shannon, que mede a similaridade entre duas distribuições de probabilidade. No nosso caso, entre a distribuição p_π e a distribuição uniforme p_u. A complexidade estatística é definida por:

$$C_{JS}[p_\pi] = Q_{JS}[p_\pi, p_u] H_S[p_\pi], \tag{6}$$

onde p_π é a distribuição de probabilidade dos padrões ordinais, p_u é a distribuição uniforme, e $H_S[p_\pi]$ é a entropia de Shannon normalizada [Borges et al. 2019].

A divergência $Q_{JS}[p_\pi, p_u]$ é dada por:

$$Q_{JS}[p_\pi, p_u] = Q_0 JS[p_\pi, p_u] = Q_0 \left\{ S\left[\frac{p_\pi + p_u}{2}\right] - \frac{S[p_\pi] + S[p_u]}{2} \right\}, \tag{7}$$

onde S é a entropia de Shannon não normalizada, e Q_0 é dada por:

$$Q_0 = -2\left\{\left(\frac{D!+1}{D!}\right)\ln(D!+1) - 2\ln(2D!) + \ln(D!)\right\}^{-1}. \tag{8}$$

Informação de Fisher A Informação de Fisher é uma medida capaz de capturar a concentração de uma dada distribuição [Borges et al. 2022]. Ela considera as diferenças entre valores de probabilidade consecutivos dentre a distribuição. Ao contrário da Entropia de Shannon, que fornece uma noção de incerteza de um sistema medindo a propagação global das distribuições, considera-se que a Informação de Fisher tem uma propriedade de localidade [Borges et al. 2022]. A Informação de Fisher é dada por:

$$F\left[p_\pi\right] = F_0 \sum_{t=1}^{D!-1} \left(\sqrt{p_{t+1}} - \sqrt{p_t}\right)^2, \tag{9}$$

onde F_0 é uma constante de normalização definida por

$$F_0 = \begin{cases} 1 & \text{se } p_{i^*} \text{ para } i^* = 1 \text{ ou } i^* = N \text{ e } p_i = 0, \forall i \neq i^*, \\ 1/2 & \text{caso contrário.} \end{cases}. \tag{10}$$

3.4. Atratores e Espaço de Fase

O espaço de fase representa o comportamento temporal de um sistema como uma trajetória pelo tempo. Atratores são um subconjunto do espaço de fase que descrevem esse comportamento [Tan et al. 2023]. Mudanças no sistema irão resultar em mudanças na série temporal e, consequentemente, na forma como o comportamento é descrito.

Quando não se conhece as equações que representam o sistema dinâmico, mas é possível observar a saída do sistema, é utilizado um método de criação de atratores denominado Coordenadas de Atraso (ou *Delay Coordinates*) [Medeiros et al. 2010]. Para a saída $s(t)$ do sistema, cada ponto x de um espaço de fase de tamanho m pode ser construído aplicando-se à função $s(t)$ atrasos τ seguidos [Medeiros et al. 2010]:

$$[x_1, x_2, x_3, \ldots, x_m] = [s(t), s(t-\tau), s(t-2\tau), \ldots, s(t-(m-1)\tau)]. \tag{11}$$

4. Análise do comportamento dinâmico do ISN

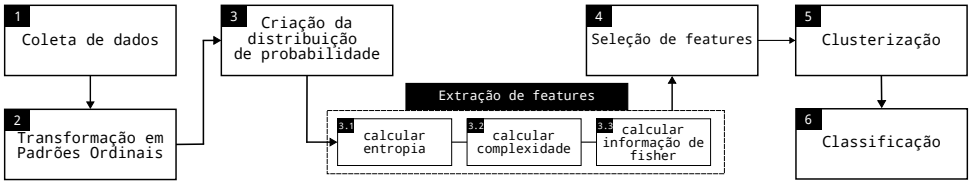

Figura 1. Etapas da análise

O processo de análise do comportamento dinâmico do ISN foi dividido em 6 etapas, ilustradas pela Figura 1. Inicialmente, (1) foram coletados 60.000 valores consecutivos de ISN de cada Sistema Operacional e equipamento analisado, em intervalos de 10

milissegundos. Em seguida, (2) os dados coletados, passam pela etapa de transformação em Padrões Ordinais, que são usados para (3) criar a distribuição de probabilidades que, por sua vez, servirá de base para a extração das *features*. A transformação em Padrões Ordinais e os cálculos são explicados em detalhes na Seção 3.

Após esta transformação, há uma etapa onde (4) são escolhidas as *features* que serão utilizadas no agrupamento e na classificação dos SOs e equipamentos. Os experimentos a respeito da seleção de *features* são detalhados nas seções seguintes. Para a etapa (5) de agrupamento, foi utilizado um método de *bootstrap* para identificar os grupos mais prováveis de acordo com seu p-valor. Isto pode ser analisado através de um dendrograma que ilustra agrupamentos em um modelo hierárquico. Por fim, na etapa (6) de classificação, verificamos a capacidade de se identificar corretamente os diferentes SOs através da análise do comportamento dinâmico de ISNs.

5. Experimentos e Resultados

Nesta seção estão apresentados os experimentos realizados neste trabalho, com o intuito de avaliar a capacidade de identificação dos diferentes SOs por meio da análise da dinâmica temporal de seu gerador de ISN. O código-fonte está disponível no github[1].

5.1. Experimento 1: Análise de Agrupamento e Seleção de Atributos

Neste experimento, buscamos identificar as diferentes classes comportamentais da evolução dos padrões ordinais de ISN ao longo do tempo em um ambiente controlado. Para isto, realizamos a coleta de ISNs de diversos switches da rede interna da UFRN-CERES (Campus Caicó). Para cada equipamento, coletamos 60.000 amostras de ISNs consecutivos, cada amostra coletada em intervalos de 10 milissegundos. A Tabela 1 apresenta informações dos fabricantes, modelos e quantidade destes equipamentos.

Tabela 1. Lista com a Descrição dos Equipamentos Switches utilizados

Fabricante	Modelo	Quantidade	Grupo	Fabricante	Modelo	Quantidade	Grupo
Aruba (AR)	2530	3	1	3Com (3C)	4210	10	3
H3C (H3C)	S5500	3	2	Hewlett Packard (HP)	1950	1	
Hewlett Packard (HP)	1910	1		Hewlett Packard (HP)	5130	6	
Hewlett Packard (HP)	1920	3		Hewlett Packard (HP)	5140	2	
Hewlett Packard (HP)	5120	6					
Total de equipamentos: 35							

5.1.1. Análise de Atratores

Em seguida, realizamos a análise dos atratores e espaço de fases das amostras coletadas. Os atratores, como apresentado na Seção 3.4, descrevem o comportamento do gerador de ISN de cada equipamento analisado. A partir da análise dos atratores de ISNs coletados dos switches, observamos a existência de apenas 3 padrões para todos os switches. A Figura 2 ilustra uma representação gráfica destes padrões identificados.

[1] `https://github.com/labepi/isn_op`

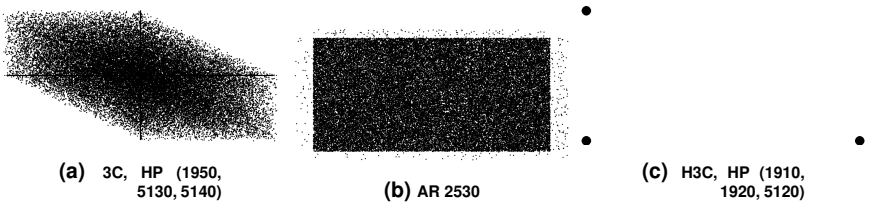

Figura 2. Representação gráfica dos padrões de atratores para os switches.

5.1.2. Análise de Padrões Ordinais

Os atratores nos dão uma representação gráfica das similaridades de geração entre estes diferentes SOs. Nesta seção, buscamos identificar a quantidade de comportamentos a partir da utilização da transformação em padrões ordinais destas sequências de ISN.

Ao realizar a transformação nas sequências de ISN, conseguimos calcular a distribuição de probabilidade de cada uma e com isso, a entropia H, a complexidade C, a informação de Fisher F. Também fizemos o mesmo processo para a diferença entre os valores da série, calculando as métricas como H', C' e F' respectivamente. Esses cálculos foram feitos para cada valor de $D \in \{2, \ldots, 7\}$.

Utilizamos o "pvclust", um pacote da linguagem de programação R que faz uma análise de agrupamento hierárquico para identificar quais são os agrupamentos mais prováveis. Essa análise é feita a partir de dois valores que são computados pela função: AU (*Approximately Unbiased*) e BP (*Bootstrap Probability*). Grupos com o valor AU maiores que 95% são destacados, indicando uma provável correlação entre os dados.

Figura 3. Agrupamento dos switches por features.

A Figura 3 apresenta os grupos identificados. Como pode-se observar, nesta análise também foram identificados apenas 3 grupos, assim como ocorreu com a visualização de atratores. Estes grupos estão identificados na última coluna da Tabela 1. Para este agrupamento, foram utilizados um total de 36 atributos. Sendo (H, C, F, H', C', F') para cada valor de $D \in \{2, \ldots, 7\}$, como mencionado anteriormente.

Em seguida, conhecendo os 3 grupos existentes neste *dataset*, realizamos a seleção das features que são mais relevantes para representá-los. Esta seleção de atributos será útil para identificarmos quais features são necessárias e suficientes para fazermos a posterior identificação dos SOs e dos equipamentos de IoT.

A Figura 4 apresenta a análise dos atributos coletados para o dataset dos switches e seu grau de importância em ordem decrescente. Esta análise considera a métrica do Incremento na Pureza do Nó (*Increase in Node Purity*) do método de seleção de importância

Tabela 2. Lista de equipamentos e Sistemas Operacionais

Label	Descrição	*Grupo*	*Label*	Descrição	*Grupo*
switch-h3c-s5500	Switch H3C		router-tplink-wr820n	Roteador TP-Link	5
switch-hp-1910	Switch HP 1910		sonicwall-2004	SonicWall 2004	
switch-hp-1920	Switch HP 1920	1	amazon-echo-dot	Amazon Echo Dot	
switch-hp-5120	Switch HP 5120		amazon-echo-pop	Amazon Echo Pop	
windows-11-pro	Windows 11 Pro		amazon-firetv-stick	Amazon Fire TV	
switch-ar-2530	Switch AR 2530		android-13-4.19.113	Android 13 (4.19.113)	
windows-10	Windows 10		android-13-5.4.274	Android 13 (5.4.274)	
windows-7	Windows 7	2	linux-2.6.32	Linux 2.6.32	
windows-8	Windows 8		linux-2.6.32-mips	Linux 2.6.32 MIPS	
windows-xp	Windows XP		linux-3.2.29	Linux 3.2.29	
freebsd-9.2	FreeBSD 9.2		linux-6.1.0	Linux 6.1.0	
ios-12.3.11	IOS 12.3.11		netbsd-5.1.2	NetBSD 5.1.2	
mac-os-x-10.8.4	Mac OS X 10.8.4		qnx-6.5	QNX 6.5	6
openbsd-4.3	OpenBSD 4.3		recalbox-6.1.77	Recalbox 6.1.77	
plan9-4	Plan9 4	3	router-ac1200	Roteador AC1200	
printer-samsung	Impress. Samsung		router-dlink-dap1320	Repetidor D-Link	
switch-3com-4210	Switch 3Com 4210		router-greatek-1200ac	Roteador Greatek	
switch-hp-1950	Switch HP 1950		router-huawei-ax2	Roteadow Huawei	
switch-hp-5130	Switch HP 5130		router-tplink-wr841n	Roteador TP-Link	
switch-hp-5140	Switch HP 5140		solaris-11.1	Solaris 11.1	
freebsd-14.1	FreeBSD 14.1		tv-lg-oled55	TV LG Oled 55"	
printer-hp	Impressora HP	4	tv-philips	TV Philips	
printer-xerox	Impressora Xerox		tv-samsung-un50	TV Samsung	

de atributos utilizando o algoritmo *Random Forest*. Esta métrica indica indica a homogeneidade nas partições das árvores com os dados passados, indicando que e a importância da variável também aumenta à medida que o valor dessa métrica aumenta.

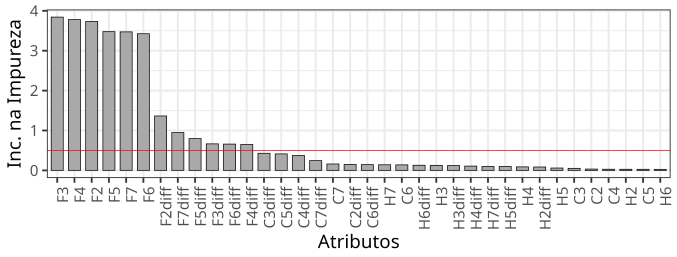

Figura 4. Análise do Incremento na Impureza dos Nós por Atributo

Desta forma, observa-se que as variáveis contendo os valores da Informação de Fisher têm um nível de importância maior do que as variáveis de entropia e complexidade. Embora os atributos de $F2$ a $F6$ se destaquem por serem mais relevantes, optamos por utilizá-las em conjunto com suas diferenças para os próximos experimentos, considerando um corte de impureza acima de 0.5, conforme linha vermelha destacada na figura.

5.2. Experimento 2: Agrupamento de Sistemas Operacionais

Este experimento tem como objetivo identificar a similaridade comportamental entre os diferentes dispositivos de IoT com diferentes SOs conhecidos. Para isso, realizamos a coleta de ISNs de diversos SOs (atuais e obsoletos) e dispositivos IoT. Além destes, incluímos uma amostra de cada um dos tipos de switches coletados no experimento anterior. Para cada sistema e dispositivo, foram coletadas uma quantidade de 60.000 amostras de ISNs consecutivos, cada uma coletada em intervalos de 10 milissegundos. A Tabela 2 apresenta informações quanto ao SO, dispositivos e a quantidade dos mesmos.

Após coletar todos os ISNs, é necessário realizar a transformação em padrões ordinais e calcular a distribuição de probabilidade, assim como a entropia H, complexidade C, informação de Fisher F e suas diferenças H', C' e F' respectivamente para cada valor de $D \in \{2, \ldots, 7\}$. Como verificado no Experimento 1, na Seção 5.1.2, a informação de

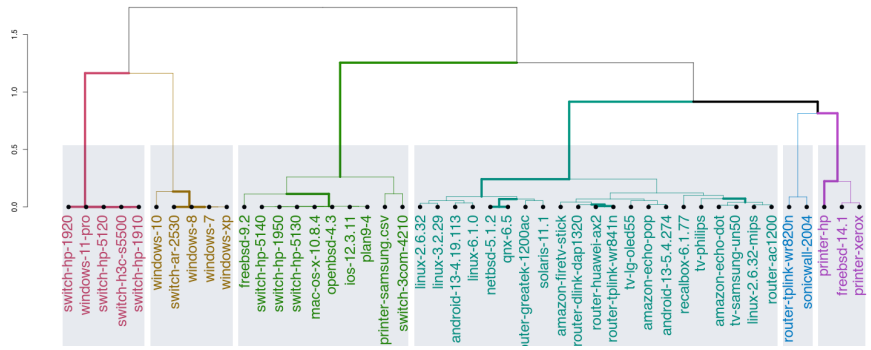

Figura 5. Agrupamento de todos os dispositivos

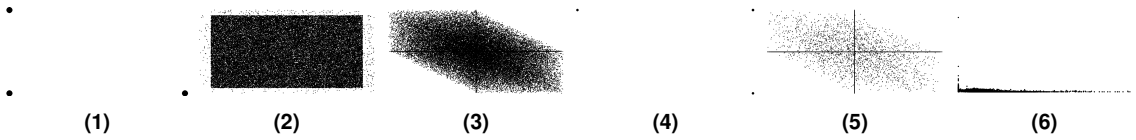

Figura 6. Representação gráfica dos padrões de atratores por grupo.

Fisher F é a *feature* mais importante dentre as calculadas. Então, para o agrupamento dos SOs e dispositivos IoT, decidimos utilizar apenas ela em conjunto com sua diferença F', totalizando assim 12 atributos.

Utilizando novamente o "pvclust", podemos visualizar o agrupamento hierárquico dos SOs e dispostivos, mostrado na Figura 5. Podemos observar também que ao invés de três grupos, como no caso do experimento com switches, foram formados seis grupos. Esse número é um possível indicador de que existem mais variantes no método de geração de ISN entre os sistemas e dispositivos.

Analisando os atratores e espaço de fases dos SOs e dispositivos, observamos a existência de seis padrões que se repetem entre os membros de seus devidos grupos determinados pela análise anterior. A Figura 6 ilustra as representações gráficas dos padrões identificados para cada grupo.

5.3. Comparação com o Nmap

É possível notar que o grupo 6 possui a maior quantidade de SOs e dispositivos comparado aos demais grupos. Um dos fatores para isso acontecer é a grande quantidade de sistemas baseados em Linux que esse grupo possui. Tendo em vista isso, podemos analisar esse grupo em específico e notar que dentro dele existem outros subgrupos. A Figura 7 ilustra os dois subgrupos encontrados.

Observa-se que os grupos na figura são divididos em dois tipos: um formado majoritariamente por SOs convencionais, e outro formado majoritariamente por dispositivos IoT. Isso demonstra a capacidade do método de distinguir até mesmo entre sistemas que possuem uma base em comum.

Uma análise foi feita com o Nmap para identificar os SO de alguns dos dispositivos IoT presentes em nossa base de dados. Utilizamos a ferramenta para verificar se ela conseguiria identificar corretamente os dispositivos, ou fazer algum tipo de distinção. A Tabela 3 apresenta os dispositivos testados e as respostas do Nmap.

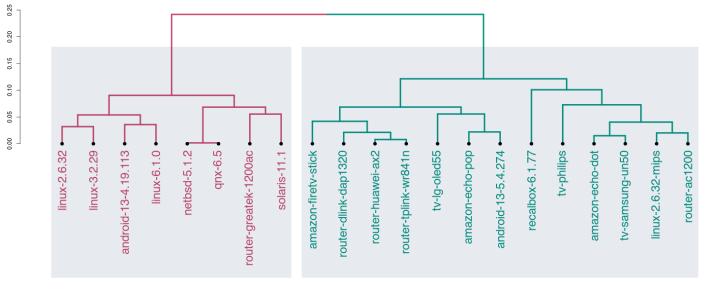

Figura 7. Análise detalhada do grupo 6

Tabela 3. Identificação remota com Nmap

Label	**Resultado Nmap**	***Label***	**Resultado Nmap**
amazon-echo-pop	Nenhuma correspondência	recalbox-6.1.77	Linux 4.X—5.X
dlink-dap1320	Linux 2.6.X	router-greatek-1200ac	Linux 3.X
router-huawei-ax2	Nenhuma correspondência	tv-philips	Linux 2.6.32 - 3.5
tv-lg-oled55	Linux 3.X—4.X	tv-samsung-un50	Muitas correspondências encontradas

É possível perceber que a maioria dos dispositivos foram identificados como Linux, provavelmente pelo fato de serem baseados nele. Com essa informação, a ferramenta não conseguiria separar bem dispositivos IoT de SOs, como foi o caso na Figura 7. Por exemplo, o Nmap identificou o dispositivo `tv-philips` como Linux 2.6.32. Já a nossa análise identificou que os dois são sistemas diferentes e as colocou em grupos distintos, quase em lados completamente opostos.

Para alguns dispositivos, o Nmap sequer achou alguma correspondência. Isso provavelmente se deve ao problema citado na Seção 1, que diz que a base de dados do Nmap é construída pela comunidade, o que pode levar à falta de dados. Outro problema semelhante acontece com o dispositivo `tv-samsung-un50` mas, dessa vez, a ferramenta encontrou muitas correspondências para a assinatura. Isso ocorre também devido a um problema citado na mesma seção, onde é necessário que as informações coletadas no momento da construção da assinatura sejam suficientes para serem comparadas com as contidas na base de dados. Uma assinatura "genérica", como foi o caso desse dispositivo, poderá ser confundida com várias outras, dificultando o processo de identificação.

5.4. Experimento 3: Classificação de SOs via Padrões Ordinais

Neste experimento de classificação, temos como objetivo avaliar a capacidade de classificação dos 46 equipamentos e SOs em seus grupos, conforme apresentados na Tabela 2, variando a quantidade de pontos coletados para cada um deles. Para isto, realizamos divisões nas séries coletadas de cada um dos dispositivos em porções menores, criando *datasets* com séries menores, mas com mais séries por dispositivo. Para isto, subdividimos o *dataset* original das série de tamanho $N = 60.000$ em cinco novos *datasets*. Como exemplo, o primeiro *dataset* possui séries com tamanho de $N = 10000$ amostras, onde, cada dispositivo tem, agora, seis diferentes séries. Desta forma, criamos mais quatro *datasets*, com séries de tamanho 5000, 2500, 1000 e 500, respectivamente. Para cada *dataset*, foram mantidos os grupos previamente identificados no agrupamento descritos na Seção 5.2 e apresentados na Tabela 2.

Com as amostras propriamente divididas, é necessário realizar a transformação dos ISNs em padrões ordinais, calcular a sua distribuição de probabilidade, extrair as métricas H, C e F, para a série original e suas diferenças H', C', F', respectivamente. Com essas informações, foi realizada uma divisão de cada dataset com 80% das amostras alocadas para o grupo de treinamento e 20% o grupo de teste. Para a classificação, utilizamos 5 algoritmos, com seus parâmetros padrão, sendo eles: *K-nearest neighbors* (KNN), com parâmetro $k = 1$, *Random Forest* (RF), com parâmetros $mtry = 7$ e $ntree = 500$, e *Support Vector Machine* (SVM) com o *kernels* Linear (SVM Linear), com $C = 1$, Radial (SVM Radial), com $\sigma \approx 3$ e $C = 1$, e, por fim, Polynomial (SVM Poly), com os parâmetros $degree = 3$, $scale = 0.1$ e $C = 1$. Os experimentos foram realizados em um ambiente de programação `R`, versão 4.4.1, e todos os classificadores foram utilizados com seus parâmetros padrão, conforme a biblioteca `caret`, em sua versão 6.0-94.

5.4.1. Resultados da Classificação

Os resultados deste experimento de classificação estão apresentados na Figura 8. Para cada classificador, foram obtidos os valores de acurácia para os cinco diferentes *datasets*, considerando os diferentes tamanhos de séries. Como pode-se observar, a quantidade de amostras coletados por série é um fator crucial para o aumento na acurácia deste método. Por exemplo, para os *datasets* com 10000 e 5000 amostras coletadas por série, o algoritmo RF atingiu uma acurácia de 100% utilizando apenas as métricas da informação de Fisher. Isso corrobora com o experimento anterior sobre a importância destes atributos.

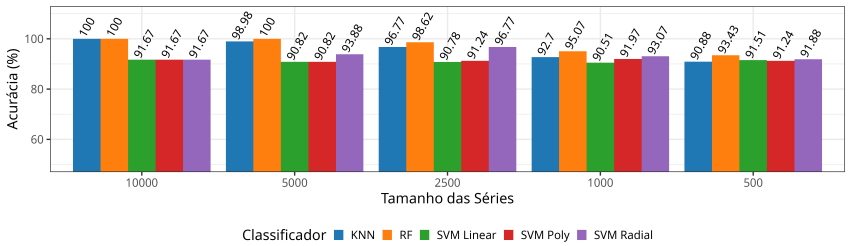

Figura 8. Resultado da Acurácia da Classificação dos Dispositivos

À medida que temos séries com menos amostras coletados por série, o resultado da acurácia dos classificadores também reduz. No entanto, observa-se que, para séries com 500 pontos apenas, o RF ainda consegue um valor de acurácia de 93.43%, o que é um valor razoável em se tratando que, nesta configuração, este *dataset* possui um tamanho de 5520 séries dentre os equipamentos comparados. Além disso, conforme descrito anteriormente, considerando um intervalo de tempo de coletas de 10ms, o tempo total para coletar os 500 pontos é de apenas 5 segundos.

Os demais classificadores obtiveram resultados maiores ou iguais a 90% de acurácia para todos os *datasets*, mantendo o comportamento de redução da acurácia à medida que menos amostras são coletadas por série. Poucas exceções podem ser observadas, como é o caso do algoritmo SVM Radial, que aumentou a acurácia de 91.67% para 93.88% e, em seguida, para 96.77%, para datasets com valores de 10000, 5000 e 2500 pontos, respectivamente, mas os valores voltam a reduzir para os demais *datasets*. Para os demais, os valores estão dentro do que se pode identificar como um estabilização

ou suave redução. Resultados estes que validam a eficácia do método, justificando a sua utilização para a identificação de diferentes SOs.

6. Conclusão

Este trabalho apresentou uma nova proposta para a identificação remota de SOs utilizando transformações em padrões ordinais, um método que ainda não tinha sido aplicado na área. Os experimentos realizados comprovam sua eficácia em reconhecer diferentes padrões de geração de ISN pelos SOs e dispositivos, como também é capaz de classificá-los com uma acurácia de 100% em certos casos. Ademais, foi possível compreender os SOs de diferentes dispositivos IoT, e o quão próximos eles estão de um SO convencional com relação à geração de ISN. Tal estudo também pode beneficiar futuras pesquisas na área de IoT e identificação remota de dispositivos inteligentes.

A principal limitação deste trabalho se dá no número de amostras N necessárias para realizar a identificação remota. Devido à natureza da transformação em padrões ordinais, é necessário satisfazer a condição $N \gg D!$, para que a transformação represente bem a série. Logo, para valores altos de D, o número total de amostras necessárias pode não ser viável. Assim, para trabalhos futuros, é necessário analisar se é viável não utilizar valores altos de D e o seu impacto nos resultados. Além disso, também podem ser analisadas outras métricas para a classificação, bem como estudar o impacto que fatores da rede podem adicionar, tais como atraso, *jitter*, entre outros. Adicionalmente, outros SOs e novos dispositivos IoT podem ser explorados e analisados, como também outros algoritmos para agrupamento e classificação.

Agradecimentos

Este trabalho foi apoiado parcialmente pela Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), processo #APQ-00426-22, CNPq, CAPES, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processos #23/00673-7 e #405940/2022-0.

Referências

Bandt, C. and Pompe, B. (2002). Permutation entropy: A natural complexity measure for time series. *Physical Review Letters*, 88(17):174102.

Beck, F., Festor, O., and Chrisment, I. (2007). *IPv6 Neighbor Discovery Protocol based OS fingerprinting*. report, INRIA. Pages: 27.

Bertino, E. and Islam, N. (2017). Botnets and IoT Security. *Computer*, 50:76–79.

Borges, J. B., Medeiros, J. P. S., Barbosa, L. P. A., Ramos, H. S., and Loureiro, A. A. F. (2023). IoT Botnet Detection Based on Anomalies of Multiscale Time Series Dynamics. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12282–12294.

Borges, J. B., Ramos, H. S., and Loureiro, A. A. F. (2022). A classification strategy for internet of things data based on the class separability analysis of time series dynamics. *ACM Transactions on Internet of Things*, 3(3):23:1–23:30.

Borges, J. B., Ramos, H. S., Mini, R. A. F., Rosso, O. A., Frery, A. C., and Loureiro, A. A. F. (2019). Learning and distinguishing time series dynamics via ordinal patterns transition graphs. *Applied Mathematics and Computation*, 362(C):1–1.

Chagas, E. T. C., Borges, J. B., and Ramos, H. S. (2022). Uso de padrões ordinais na caracterização e análise de ataques de botnets em internet das coisas (iot). In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, page 133–137. SBC.

Eddy, W. (2022). Transmission Control Protocol (TCP). RFC 9293.

Fan, H., Kong, B., Li, G., Li, J., Zhang, J., An, Y., Wan, J., Zhang, Z., and Fan, J. (2022). A random forest-based operating system recognition algorithm for network security. In *2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, pages 539–550.

Feldman, D. P. and Crutchfield, J. P. (1998). Measures of statistical complexity: Why? *Physics Letters A*, 238(4–5):244–252.

Gont, F. and Bellovin, S. (2012). Defending against Seq. Number Attacks. RFC 6528.

Laštovička, M., Husák, M., Velan, P., Jirsík, T., and Čeleda, P. (2023). Passive operating system fingerprinting revisited: Evaluation and current challenges. *Computer Networks*, 229:109782.

Lyon, G. F. (2009). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure, Sunnyvale, CA, USA.

Medeiros, J. P. S., Brito, A. M., and Motta Pires, P. S. (2010). An effective tcp/ip fingerprinting technique based on strange attractors classification. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 208–221, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ordorica, A. (2017). Operating System Identification by IPv6 Communication using Machine Learning Ensembles. *Graduate Theses and Dissertations*.

Rosso, O. A., Larrondo, H. A., Martin, M. T., Plastino, A., and Fuentes, M. A. (2007). Distinguishing noise from chaos. *Physical Review Letters*, 99(15):154102.

Shamsi, Z., Nandwani, A., Leonard, D., and Loguinov, D. (2016). Hershel: Single-packet os fingerprinting. *IEEE/ACM Transactions on Networking*, 24(4):2196–2209.

Song, J., Cho, C., and Won, Y. (2019). Analysis of operating system identification via fingerprinting and machine learning. *Computers & Electrical Engineering*, 78:1–10.

Tan, E., Algar, S. D., Corrêa, D., Stemler, T., and Small, M. (2023). Network representations of attractors for change point detection. *Communications Physics*, 6(1):1–14.

Zanin, M. (2023). Continuous ordinal patterns: Creating a bridge between ordinal analysis and deep learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(3):033114.