

Characterising Network-Connected Devices Using Affiliation Graphs

Kyle Millar*, Adriel Cheng^{*,†}, Hong Gunn Chew*, and Cheng-Chew Lim*

^{*}School of Electrical and Electronic Engineering, The University of Adelaide, Australia

[†]Cyber and Electronic Warfare Division, Defence Science & Technology Group, Australia

Email: {kyle.millar, adriel.cheng, honggunn.chew, cheng.lim}@adelaide.edu.au*

Email: adriel.cheng@dst.defence.gov.au[†]

Abstract—Device management in large networks is of growing importance to network administrators and security analysts alike. The composition of devices on a network can help forecast future traffic demand as well as identify devices that may pose a security risk. However, the sheer number and diversity of devices that comprise most modern networks have vastly increased the management complexity. Motivated by a need for an encryption-invariant device management strategy, we use affiliation graphs to develop a methodology that reveals key insights into the devices acting on a network using only the source and destination IP addresses. Through an empirical analysis of the devices on a university campus network, we provide an example methodology to infer a device’s characteristics (e.g., operating system) through the services it communicates with via the Internet.

Index Terms—Affiliation graphs, device discovery and management, passive network reconnaissance.

I. INTRODUCTION

Device management aims to determine what devices are acting on a network and the purpose of their use. This process helps network administrators provide better quality of service (QoS) and allows security analysts to locate vulnerable devices within their network [3]. However, with the growing number and diversity of devices, implementing an effective device management strategy is increasingly difficult in practice.

A typical enterprise network (such as a university network) is likely to consist of tens of thousands of devices. These devices range from the servers critical to the overall operation of the network, to the personal devices of each user. Additionally, the Internet of things (IoT) paradigm has led to a rapid influx of heterogeneous devices present on such networks (e.g., printers, IP cameras, and smart sensors) [11]. The rising complexity of providing adequate device management has not only hindered the administration of enterprise networks but has also introduced security vulnerabilities due to unknown and unpatched devices [2]. It is therefore evident that methods of reducing the complexity of device management is of critical importance.

Current techniques of device management have typically relied on either a large number of difficult to extract fea-

tures (e.g., Internet protocol (IP) ID monotonicity and clock frequency) or target a specific subset of devices [6]. The wide array of devices present on modern networks renders these techniques either computationally ineffective at scale or are not universally applicable. It becomes necessary to infer information about the devices on a network through a limited but universally applicable feature set.

Affiliation graphs are a social networking technique that investigates the characteristics of an individual through the communities they are affiliated with [5]. Social communities form through both implicit and explicit commonalities in their membership set; such as shared interests, occupation, or genealogy [14]. Through analysing the shared commonality within a community, certain characteristics of its membership set can often be revealed. For example, Facebook is able to identify the political alignment of a user based solely on the Facebook groups the user is affiliated with [9].

The advantage of an affiliation graph lies in the ease in which the affiliations between an individual and their communities can be observed [4]. Through understanding why these affiliations are made, interesting inferences can be made about an individual without requiring any *a priori* knowledge of the individual themselves.

It is the axiom of this work that certain device characteristics (e.g., manufacturer, operating system (OS), and installed applications) can be determined by the communities a device affiliates with via the Internet. For example, a device affiliated with a Windows update server is likely to be running the Windows OS [6]. Through extrapolating this approach to additional communities with known commonalities, a profile of the device’s overall characteristics could therefore be constructed.

Investigating the communities that a device is affiliated with rather than the device itself is useful for two key reasons. First, the characteristics of an unknown set of devices can be inferred from a known set of devices given they exhibit a similar structural equivalence.¹ Second, it allows for the identification of a device’s characteristics without requiring direct analysis of the device itself. This is useful in various scenarios such as when the device is only intermittently connected to the

This research is supported by the Australian Government Research Training Program (RTP) Scholarship and the Commonwealth of Australia as represented by the Defence Science & Technology Group of the Department of Defence.

¹In an affiliation graph, the structural equivalence of two devices is proportional to the number of communities they share.

network (e.g., a mobile device) or the device is no longer present on the network at all.

In our study, we examine the relationship between devices on a network (*actors*) and the destination addresses (*communities*) in which these devices communicate with via the Internet. This simplifies our analysis to just two easily obtainable fields – the source and destination IP addresses. In restricting our analysis to just these two fields, we create an encryption invariant basis of analysis that is quick to populate and requires little storage. An example of constructing an affiliation graph from the source and destination IP addresses is shown in Figure 1.

The key contribution of our work is twofold: (1) To the best of our knowledge, this is the first work which identifies a device’s characteristics through analysis of the destination IP addresses (*communities*) it communicates with. The examination of a device’s affiliated communities, rather than the device itself, provides a method of device characterisation that is useful when the device itself cannot be examined; and (2) we provide an empirical analysis of our technique on a university campus network. In particular, we provide an example of how this method could be implemented to determine a device’s OS.

The remainder of this paper is structured as follows: A brief survey of related work in device management is given in Section II. Section III outlines the process of creating an affiliation graph from two 26-hour captures of a university’s network. Analysis of an affiliation graph’s community set in relation to network traffic is explored in Section IV. Section V provides an empirical example of how the analysis of this

community set can give insight into the operating systems of the devices on a network. The paper concludes in Section VI with a discussion of results and suggestions for future work.

II. RELATED WORK

There has been growing research interest in device management strategies due to the increase in the number of devices with Internet connectivity; however, device management is still underrepresented by the research community at large.

Arora et al. [3] highlighted the security benefits of identifying key devices on a network. The knowledge of these key devices and their typical characteristics can help to assess and mitigate the impact in the event of a security breach.

Arora et al. went on to investigate techniques to identify the servers present on a network. Although achieving high detection rates using machine learning techniques, several feature extraction stages were first required. The computational requirement of these feature extraction stages reduces the real word efficacy of this method for any moderate sized network. Furthermore, the identification of servers alone is insufficient to effectively administer a network. Increasingly, bring-your-own device (BYOD) policies are causing issues in the management of enterprise networks [13].

BYOD policies have been widely introduced in educational and business networks such that their users can connect their personal devices to the network. While this is convenient for users of the network, it introduces novel challenges in how to manage and secure a ‘user device’ for which the network administrators have little control over.

IoT devices have also posed serious concerns for user privacy and security within an enterprise network [11]. Methods to mitigate these concerns have sparked interest in device management strategies to identify and manage IoT devices operating on a network [8]; however, the current state of IoT device management has been complicated due to the heterogeneity and rapid evolution of such devices.

In this paper, we introduce a novel method of device characterisation through the use of affiliation graphs. Although there has been speculation that a device’s characteristics can be inferred through the services for which it connects to via the Internet [6], [13], this paper provides a methodology to extract these relationships. In particular, our method aims to address the dynamic nature of device characterisation by utilising only two universally applicable features – the source and destination IP addresses.

III. AFFILIATION GRAPHS FOR DEVICE MANAGEMENT

The axiom of affiliation graphs is that there exists a reason for which actors are affiliated with certain communities [14]. For example, Internet-connected devices will either contact or be contacted by certain servers to synchronise the services running on the device (e.g., Facebook notifications). This insight suggests that the client-side services of a device could be inferred through their affiliation to their serve-side counterpart.

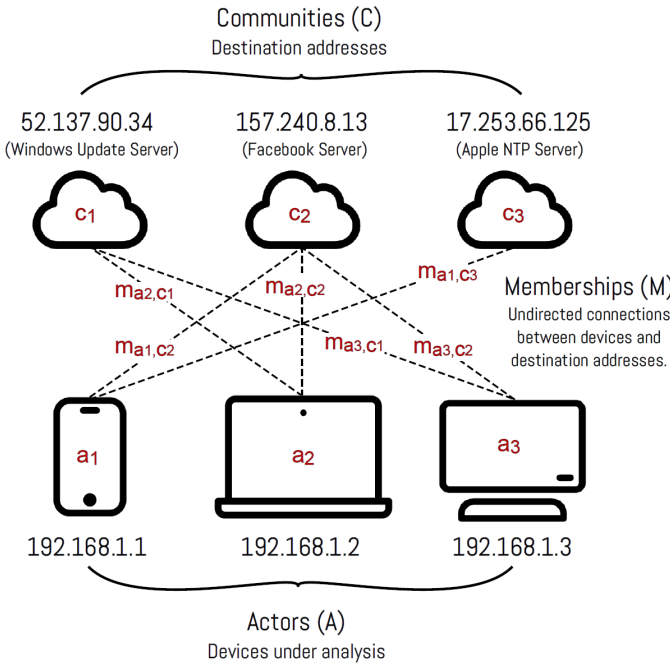


Fig. 1. An affiliation graph created from network data. The actors and communities of the affiliation graph are the devices internal and external to the network under analysis, respectively. The membership set consists of the network communications between the internal and external devices.

A. Notation

An affiliation graph is a bipartite graph consisting of two distinct sets of vertices - actors (A) and communities (C) [5]. An actor ($a \in A$) is said to be affiliated with a community ($c \in C$) if there exists a membership ($m_{a,c} \in M$) between a and c ; where M is the set of all memberships between A and C . An affiliation graph can therefore be represented as the collection of actor, community, and membership sets, $G = (A, C, M)$.

B. Populating an Affiliation Graph from Network Data

The benefit of an affiliation graph is the ease in which it can be constructed. To create an affiliation graph from network data, only the source and destination IP addresses are required from each network flow². This results in a technique that is quick to populate and requires little storage. Additionally, this technique is also widely deployable as most network administrators typically have access to such flow records.

The pseudo-code for creating an affiliation graph from network data is provided in Algorithm 1. It should be noted that this method utilises a weighted affiliation graph such that the weight of a membership, $\omega(m)$, is the number of flows between a device and a community.

C. University Campus Data Set

To analyse the efficacy of applying affiliation graphs on a large and diverse network, two captures of the University of Adelaide's enterprise network were taken. Both captures consisted of 26 hours of network traffic for which all flow records were recorded. The first capture, named *low-activity*, was taken on Christmas day in 2018 and captured approximately 19 million flow records. This capture was scheduled to ensure that as few devices were active on the network as possible and therefore serves as a baseline profile of the network. The second capture, named *high-activity*, was taken on the 1st of May 2019³ and

captured approximately 33 million flow records. This capture was taken during a typical workday where the network was actively being used by both staff and students.

The *low-activity* affiliation graph took approximately half an hour to populate; whereas, the *high-activity* affiliation graph took just under an hour. The time taken to create both affiliation graphs highlights the ability of this method to scale well even when the activity on a network increases significantly. Table I provides a summary of the affiliation graphs created from the two network captures. From this table, the following insights were identified:

1. The number of devices present on the *high-activity* capture is significantly greater than that of the *low-activity* capture. This result is consistent with the increase in users on the network during the *high-activity* capture.
2. Less than 15% of flows within both captures have a unique set of source and destination IP addresses. This result shows that the majority of traffic generated or received by a device consists of repeated connections to certain communities.

A device's affiliation to a given community can give insight into the device itself. For example, the devices affiliated with the community '210.173.216.59' within the *high-activity* affiliation graph are depicted in Figure 2. This community is owned by 'Ricoh Company, Ltd.', a company most notably known for manufacturing cameras and printers. It was found that of the 90 devices affiliated with this community, 89 were printers manufactured by the Ricoh company. It is therefore likely that an additional device (new or previously unknown) which affiliates with this community would also be a Ricoh printer. This example illustrates the ability to draw information about a device through their affiliated communities. In the next section, techniques to extract useful information about the community set are examined.

IV. COMMUNITY ANALYSIS

A unique property of an affiliation graph created from network data is the wealth of information that can be gathered about the community set. Through utilising standard network reconnaissance tools such as WHOIS [7] and the reverse domain name system (rDNS) [10], the service that a community provides can often be determined. This examination of the community set can provide a useful technique of inferring information about a device even when the examination of the device itself is infeasible (e.g., when the device is no longer active on the network under analysis).

TABLE I

AFFILIATION GRAPHS SUMMARY STATISTICS. $|A|$ IS THE NUMBER OF DEVICES, $\mu_{deg(A)}$ IS THE MEAN NUMBER OF COMMUNITIES PER DEVICE, $|C|$ IS THE NUMBER OF COMMUNITIES, $\mu_{deg(C)}$ IS THE MEAN NUMBER OF DEVICES IN EACH COMMUNITY, AND $|M|$ IS THE NUMBER OF MEMBERSHIPS.

Data Set	$ A $	$\mu_{deg(A)}$	$ C $	$\mu_{deg(C)}$	$ M $
<i>low-activity</i>	6.1k	201	476k	2.55	1.22M
<i>high-activity</i>	24.7k	158	566k	6.89	3.90M

Algorithm 1 Populating an Affiliation Graph

```

1: for flow in flow records do
2:   Record source and destination IP of flow.
3:   Set device,  $a$ , as the IP address within the network
   under analysis.
4:   Set community,  $c$ , as the IP address outside the
   network under analysis.
5:   Denote the membership between  $a$  and  $c$  as  $m_{a,c}$ .
6:   If  $a \notin A$ ,  $A = A \cup \{a\}$ 
7:   If  $c \notin C$ ,  $C = C \cup \{c\}$ 
8:   If  $m \notin M$ ;  $\omega(m_{a,c}) = 1$ ,  $M = M \cup \{m_{a,c}\}$ ; else,
    $\omega(m_{a,c}) = \omega(m_{a,c}) + 1$ 
9: end for

```

where $\omega(m)$ is the weight of the membership m .

²A network flow (or flow) is defined in this work as the bi-directional exchange of information between two devices over the Internet.

³The 1st of May is not a public holiday in the location in which the capture took place.

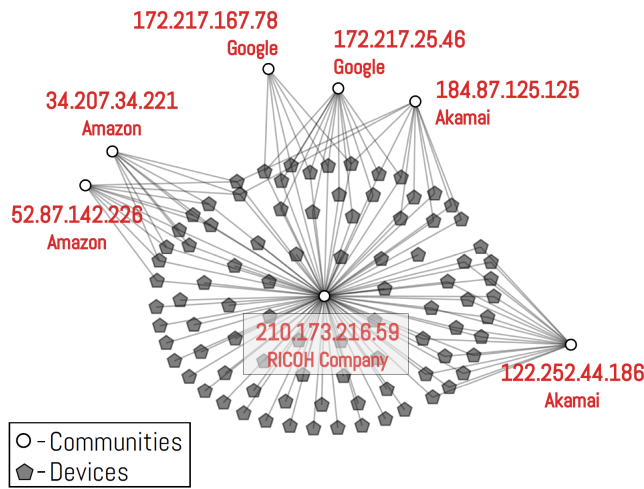


Fig. 2. A subset of the *high-activity* affiliation graph. This subset depicts the devices affiliated with the community ‘210.173.216.59’. This community is owned by ‘Ricoh Company, Ltd.’. 89 of the 90 devices affiliated with this community were found to be printers manufactured by Ricoh itself.

It is worth noting that the analysis of the community set can be done independently from the analysis of the target network itself. This allows community analysis to be evaluated in an environment for which the examiner has complete control.

The service a community provides can give insight into the reason for which a device is affiliated with a certain community. For example, the community ‘157.240.8.35’ can be resolved to one of Facebook’s Australian servers. A device with a membership to this community is therefore likely (1) located in Australia, (2) running a Facebook service, and (3) a user device. Additionally, the device is also likely to be a mobile device as it has been found that 79% of social media is now accessed through mobile devices [1]. A profile of the device’s overall characteristics (e.g., manufacturer, OS, and installed applications) could then be created through extrapolating this approach to additional known communities.

To investigate what device characteristics could be inferred from community properties; the (1) ownership, (2) domain name, and (3) associated application traffic of each community were resolved where possible. From these community properties, the service a community provides could then be estimated.

1) Ownership — The ownership of an IP address is the registered entity responsible for that IP address. The ownership of a community can provide an overview of the types of services the community provides. For example, a community owned by Facebook will likely host one of Facebook’s social networking services (i.e., Whatsapp, Instagram, and Facebook). While the exact service of the community cannot be determined in this scenario, it reduces the search space for the potential services that are likely to be offered by this community. Additionally, in the case where the registered owner only provides a singular service (e.g., Twitter), the ownership of an IP address can directly signify the purpose of its respective community.

The ownership of an IP address can be resolved through a WHOIS query. A WHOIS query will return information about

the registered entity responsible for the queried IP address. This information can then be used to determine what services are likely to be associated with this community.

The ownership of a community however is not always useful for the purposes of our analysis. For example, content distribution networks (CDNs) provide many distinct services under a singular ownership. Therefore, further information about such communities would be required to determine their services.

2) Domain Name — A device’s domain name is a human readable name associated with the device. Often a device’s domain name is created using a particular naming convention which can provide useful hints to the service the device provides. For example, the domain name of the community ‘17.253.66.125’ was resolved to *ausyd2-ntp-001.aaplimg.com* using rDNS. This domain name suggests that the community is (1) located in Sydney, Australia (*ausyd*); (2) a network time protocol (NTP) server (*ntp*); and (3) owned by Apple (*aaplimg.com*⁴). Therefore, the service of this community is likely to be providing accurate time resolution to Apple devices within the Australia region. It should be noted however that the domain name of a community within a CDN is unlikely to resolve further information about the community itself.

The main purpose of a CDN is to provide distributed hosting of popular Internet applications (e.g., YouTube, iTunes, and Twitter) [12]. The service provided by a CDN community is therefore closely associated with the application it hosts. The service of a community CDN can therefore often be inferred through analysis of its associated application network traffic.

3) Application Traffic — The application traffic associated with a given community can be a strong indication of the service that community provides. For example, 97% of the traffic to and from the community ‘74.125.130.109’ was found to be associated with Google’s mail application, Gmail. Therefore, it can be assumed that a device affiliated with this community is using the Gmail application.

The application traffic associated with each community was identified through correlating the application logs taken during the capture period. These application logs were generated through a combination of deep packet inspection and metadata analysis. Resolving the application traffic associated with a community can only be achieved when there exists a system of classifying application traffic on the network; however, once the service of a community is identified it can be used in subsequent analysis without requiring its properties to be resolved again.

An example of resolving all three community properties for three example communities is given in Table II. It is seen that multiple correlating sources of information allow for a strong indication of the service that a community offers.

The community properties resolved within this section are likely to be the reason for which a device would affiliate with

⁴The ownership of the *aaplimg.com* domain is owned by Apple, Inc. as resolved through a WHOIS query

TABLE II
IDENTIFIED PROPERTIES OF THREE EXAMPLE COMMUNITIES.

Community Example	IP Address	Ownership	Domain Name	Application Traffic	# low-activity devices	# high-activity devices
#1	157.240.8.35	Facebook, Inc.	edge-star-mini-shv-01-syd2.facebook.com	Facebook (95%), Incomplete (3%), SSL (2%)	322	9,799
#2	17.253.66.125	Apple, Inc.	ausyd2-ntp-001.aaplimg.com	NTP (100%)	235	1,617
#3	74.125.130.109	Google, Inc.	sb-in-f109.1e100.net	Gmail (97%), Incomplete (3%)	56	2,389

a specific set of communities. For example, a device affiliated with the Facebook community in Table II is presumably running the Facebook client-side application. The utilisation of these inferences allows for information about the set of devices to be resolved through analysing the communities they affiliate with.

V. DEVICE ANALYSIS

This section serves as a preliminary example of how the community analysis in Section IV can be applied to infer the characteristics of the devices operating on a network. The example provided in this section examines the use of an affiliation graph to identify the operating systems (OSs) of a device. Identifying the OS of the devices on a network is critical for providing quality of service and assessing the security vulnerabilities of the network as a whole. As such, OS classification is an important area of interest in device characterisation.

The axiom of identifying the OS of a device through an affiliation graph is that there is an intrinsic relationship between the device's OS and the communities it affiliates with. Intuitively, this axiom seems plausible given a device will often connect to services related to its operating system (e.g., to check for new notifications and software updates). This intuition is illustrated in Figure 3 where the community affiliations of eight devices in the *high-activity* capture were examined. It is seen that a large portion of the affiliations of each device are to the maintainers of their respective OS.

To investigate the use of an affiliation graph to distinguish between distinct OSs, a subset of known devices from the high-activity capture were analysed. In total, 10,521 devices were selected for analysis as their operating systems could be validated through either auxiliary logs or via physical inspection. The distribution of operating systems under analysis in this section is shown in Table III.

The following subsections investigate whether a particular OS can be inferred through only evaluating its affiliated communities. A summary of the results found for Apple OS and Windows OS can be seen in Figure 4.

A. Apple OS

A strong relationship was found between devices running an Apple OS⁵ and the communities owned by Apple itself.

⁵Apple OS has been defined as either Apple's mobile OS (iOS) or desktop OS (OS X).

Given the set of devices running an Apple OS, 5,239 (95.7%) were affiliated with five or more Apple owned communities. Conversely, of the 5,285 devices that affiliated with five or more Apple communities, 99.1% were a device running an Apple OS. This result indicates a significant dependency on Apple's online ecosystem to manage the client-side services running on Apple's OSs. Furthermore, due to Apple's exclusivity in offering services for only its own products, it proves trivial to separate an Apple OS from other OSs through only resolving the ownership of its affiliated communities⁶. Further investigation however would be required to distinguish between devices running iOS and OS X using this method.

B. Windows OS

Through investigating the ownership of the communities alone there proved to be no significant relationship between communities owned by Microsoft and devices running Windows OS. This result is not unexpected as Microsoft provides a wide

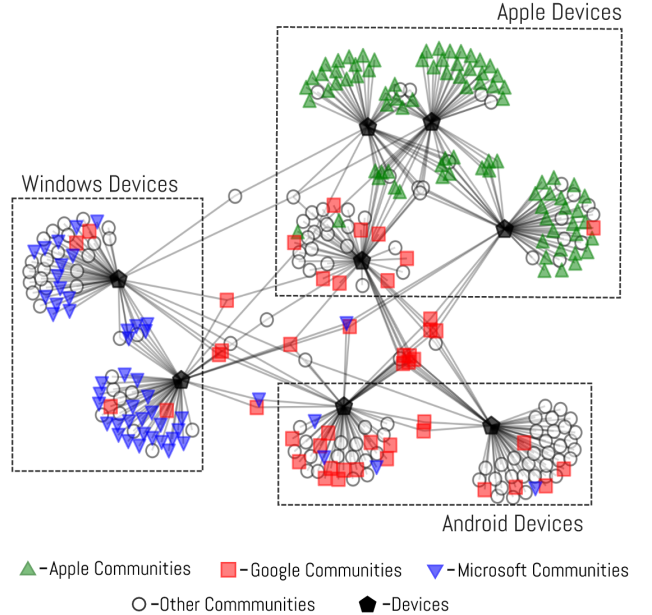


Fig. 3. A subset of the *high-activity* affiliation graph depicting the community affiliations of eight devices with known characteristics. Four devices which were running an Apple OS (three iOS, one OS X), two devices running Windows OS, and two devices running the android OS. For clarity, only the top 50 most heavily weighted community affiliation of each device is depicted.

⁶provided the device is exhibiting typical network behaviour.

TABLE III
DISTRIBUTION OF OPERATING SYSTEMS IN THE KNOWN SUBSET OF THE
HIGH-ACTIVITY AFFILIATION GRAPH.

Operating System	Number of devices
iOS	3,872 (36.80 %)
OS X	1,602 (15.23 %)
Windows	3,285 (31.22 %)
Android	1,762 (16.75 %)
Total	10,521 (100.0 %)

range of services that are not only limited to devices running Windows OS (e.g., Office365). Therefore, a greater focus was placed on the service the community provides rather than its ownership alone.

From the analysis conducted in Section IV, 122 communities offering Windows specific services were found – 111 providing Windows updates and 11 providing the Windows push notification service (WNS). Of the 1,101 devices with an affiliation to at least one of these communities, 2,454 (97.3%) were confirmed to run Windows OS. Additionally, 74.7% of the 3,285 devices running a Windows OS were identified to affiliate with one or more of these communities. This result indicates that there is a relationship between communities offering Windows related services and devices running Windows OS; however, it is not as comprehensive as the relationship between Apple OS and Apple based communities.

C. Android OS

No distinct set of Google communities were found to result in a significant relationship with devices running Android OS. This result is due to the myriad of OS-agnostic services that offered by Google (e.g., Google's search engine, Chrome, and YouTube). Furthermore, as Android is maintained by many distinct entities it proves more difficult to determine a specific set of communities that would be representative of its online services.

A potential method to increase the information contained in the affiliation graph would be to differentiate the community set not only by an IP address but also by port number. This would allow the analysis to distinguish between distinct services running on the same IP address.

VI. CONCLUSION AND FUTURE WORK

In this paper, the application of affiliation graphs for characterising devices on an enterprise network was evaluated. A methodology for analysing the community set was investigated which allowed for inferences to be made about the characteristics of a device through the communities it affiliates with. An example was given to demonstrate the applicability and effectiveness of applying an affiliation graph to identify the OSs of the devices on a network. Overall, it was found that an affiliation graph is a quick and easy method to gain an initial insight into the characteristics of the devices on a network. However, further analysis should be conducted to identify robust relationships between devices characteristics and their affiliated communities.

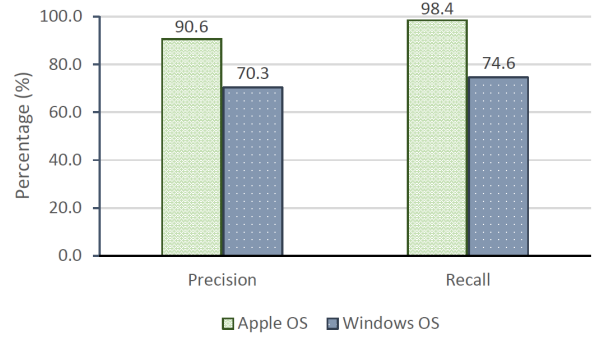


Fig. 4. Precision and recall of Apple and Windows devices found through investigating their community set. Apple devices are defined as a device running either iOS or Mac OS; whereas, Windows devices are defined as a device running the Windows OS.

In future work we aim to provide a broader analysis on utilising affiliation graphs for device characterisation. In particular, we aim to utilise supervised machine learning techniques to help identify the relationship between a device's characteristics and the communities it affiliates with.

REFERENCES

- [1] The global mobile report. <https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/The-Global-Mobile-Report>, 2017. [Online] Accessed on: September 2019.
- [2] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1093–1110, Vancouver, BC, 2017. USENIX Association.
- [3] Deepali Arora, Kin Fun Li, and Alex Loffler. Big data analytics for classification of network enabled devices. In *Advanced Information Networking and Applications Workshops (WAINA)*, pages 708–713, 2016.
- [4] Stephen Borgatti and Daniel Halgin. Analyzing affiliation networks. *The Sage Handbook of Social Network Analysis*, 2011.
- [5] Ronald L Breiger. The duality of persons and groups. *Social Forces*, 53(2):181–190, 1974.
- [6] Yi-Chao Chen, Yong Liao, Mario Baldi, Sung-Ju Lee, and Lili Qiu. OS fingerprinting and tethering detection in mobile networks. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 173–180. ACM, 2014.
- [7] Leslie Daigle. RFC-3912 WHOIS protocol specification. *Network Working Group*, 2004.
- [8] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martin Ochoa, Nils Ole Tippenhauer, Juan Davis Guarnizo, and Yuval Elovici. Detection of unauthorized IoT devices using machine learning techniques. *arXiv preprint arXiv:1709.04647*, 2017.
- [9] Jeremy Merrill. Liberal, moderate or conservative? see how Facebook labels you. <https://www.nytimes.com/2016/08/24/us/politics/facebook-ads-politics.html>, 2016. [Online] Accessed on: September 2019.
- [10] Paul Mockapetris. RFC-1035 domain names-implementation and specification. *Network Working Group*, page 23, 1987.
- [11] Bruce Schneier. The internet of things will upend our industry. *IEEE Security & Privacy*, 15(2):108–108, 2017.
- [12] Athena Vakali and George Pallis. Content delivery networks: status and trends. *IEEE Internet Computing*, 7(6):68–74, Nov 2003.
- [13] Xuetao Wei, Nicholas C Valler, Harsha V Madhyastha, Iulian Neamtii, and Michalis Faloutsos. Characterizing the behavior of handheld devices and its implications. *Computer Networks*, 114:1–12, 2017.
- [14] Jaewon Yang and Jure Leskovec. Community-affiliation graph model for overlapping network community detection. In *IEEE 12th International Conference on Data Mining*, pages 1170–1175. IEEE, 2012.