Knowledge Discovery of Port Scans from Darknet

Sofiane Lagraa, and Jérome François
Email: sofiane.lagraa@inria.fr, jerome.francois@inria.fr
INRIA Nancy-Grand Est, 615 rue du Jardin Botanique, 54600 Villers-les-Nancy, France

***Abstract*—Port scanning is widely used in Internet prior for attacks in order to identify accessible and potentially vulnerable hosts. In this work, we propose an approach that allows to discover port scanning behavior patterns and group properties of port scans. This approach is based on graph modelling and graph mining. It provides to security analysts relevant information of what services are jointly targeted, and the relationship of the scanned ports. This is helpful to assess the skills and strategy of the attacker. We applied our method to data collected from a large darknet data, i.e. a full /20 network where no machines or services are or have been hosted to study scanning activities.**

I. Introduction

Computers connected to a network use many services by mainly relying TCP/UDP protocols. Port or IP scanning (also known as sweeping) is one of the most common techniques by attackers to discover open ports in preamble of an attacker or an intrusion through those ones. Hence, scanning methods are part of network-based discovery techniques, still for emerging threats like Advanced Persistent Threats [5]. Therefore, an in-depth understanding of scanning techniques is necessary for improving security: detection, prevention or forensics. There are three main types of scans: vertical, horizontal and block scans. Vertical scan is described as a single IP being tested on multiple ports. Horizontal scan is described as trying scan against a group of IPs for a single port. Block scan is a combination of both of them. All of them cannot.

This paper aims at making a deeper and more sophisticated analysis of vertical scans by seeking the relationship of commonly scanned TCP ports. Our approach relies on building dependencies among those latter and extract the predominant roles of certain ports within sequence of consecutive targeted ports as well as extracting relationships among them, *i.e.* discover groups of commonly scanned ports. This can help the security analysts to guard against attacks and improve prevention and detection tools. For example, given the scans performed on the following successive ports 80, 591, 8008, 8080 and 443, we want to discover the relationship of commonly scanned ports with their dependencies, then the significant service ecosystem is that these ports are used for "HTTP" traffic. Thus, the analysts can update the web server security. We have validated the method on real data collected from a darknet in order to observe and understand (1) the behavior of different scans and (2) extract the "modi operandi" of targeted ports.

In this paper, our contributions are as follows:

1) a graph-based model of port scans relationships,
2) a knowledge discovery methodology using graph mining techniques based on the proposed model,
3) an application of our methodology on real darknet data.

The rest of the paper is organized as follows. Section II briefly presents the background on darknet and related work of analysis techniques of ports scans. Section III provides an overview of dataset and highlights research questions. Section IV and Section V present the first and the second part of our approach concerning the port-scan graph model and port-scan graph analytic, respectively. Finally, Section VI draws conclusion and discuses the future work.

II. Background and Related Work

A vertical port scanner sends a message to each port of predefined list and waits for a certain replies. For example, a common manner (TCP SYN-scan) to discover TCP open ports is to try to initiate new connections with SYN-flagged packets (TCP SYN scan). If the port is open, a SYN/ACK packet is sent back while a RST packet is representative of a closed port.

From darknet data we have collected, we conduct the study over all TCP-SYN packets which represent 93,79% of TCP packets. The darknet (network telescopes) refers to the servers configured to trap adversaries and collect suspicious data. Since these servers run without interacting with no legitimate users (attackers) in passive mode and correspond to unused hosts or devices, any observed traffic destined to them raises suspicion and hence necessitates investigation. They do not offer Internet services, nor does it use any. Darknet has been used in the past to extract different insights on probes or scanning activities [3], [10], bots, DDoS (Distributed Denial-of-Service) attacks due to victims reply (backscatter) packets to spoofed IP addresses [14]. For instance, DDoS attacks and scanning activities can be detected using the existing techniques and tools such as IDS Suricata[1], topological analysis technique [7], and statistical techniques [9]. However, these tools and techniques provide scanning profiles of ports such as the number of countries are responsible for the majority of large scans, the number of performed scans, and the targeted services.

Darknets are also known as darkspace, blackhole monitors, unused IP addresses, or network telescopes. To harmonize the terminology, we use the word darknet throughout this paper. A darknet is a whole subnetwork, which is announced over Internet such that packets sent to the IP addresses are properly routed over. This subnetwork is defined by the prefix length. It does not host any services and so no legitimate traffic is

[1] https://suricata-ids.org/

978-3-901882-89-0 @2017 IFIP

Protocol	Number of packets	Weight (%)
TCP	2,265,756,934	78.54%
UDP	352,449,519	12.21%
ICMP	33,235,987	1.15%

TABLE I: Number of packets per protocol

supposed to reach it. The entity hosting the darknet is then silently collecting all incoming packets, i.e. without replying to any of them. They have been proved to be appropriate monitoring techniques for security, e.g. to observe botnets, scans or distributed denial-of-service attacks [3], [10], [1]. Although there can be primarily seen as very naïve collector, especially when compared to more advanced Internet security monitoring approaches like honeypots, they have been demonstrated to be complementary [11]. In [1], the authors discovered patterns of darknet traffic and how it changes over time.

In [8], the authors presented the measurement and analysis of a 12-day world-wide cyber scanning campaign targeting VoIP (SIP) servers using a darknet. Similarly, in [9], behavior of horizontal scanning scans have been researched on. In [18], the authors presented an animated 3-D scatter plot visualization of port scanning on darknet data. Finally, in [7], a topological data analysis technique is able to analyze and visualize a large number of IP packets in order to make malicious activities patterns easily observable by security analysts. Among these patterns, they found scanning activities observed by a darknet.

Compared to previous approaches, it is worth to mention that our work focuses on vertical scan. Besides, most of current studies either rely on evaluating trends over targeted ports or behavior of a scan towards a single ports. Our approach tends to automatically discover dependencies among targeted ports rather than deducing those relations thanks to observation like in [8]. In that sense it is closer to work related about automated discovery of network service dependencies [17]. However, the goal of the latter is to discover normal dependencies that exist in benign applications whereas ours establish relations in attacker behaviors when performing TCP scans. To achieve that, we provide behavior pattern of port-scan through graph mining methods which provide intrinsic analysis of scanned ports where the classical statistical tools do not allow to discover.

III. Dataset overview and research questions

Darknet data used in this paper are collected from a /20 darknet network (i.e., 4096 addresses) for 2 years (Nov.2014 $\sim$ Nov2016). During this period 2,884,539,435 packets were captured representing 500 GB of data with an increasing trend over the months as shown in Figure 1. As highlighted in Table I, the number of TCP packets represents more than 78% of packets. This is a challenging task to analyze port-scan strategies from a huge amount of traffics.

Since, there are 65,536 possible TCP ports, we segment the scanned port numbers on three ranges according to Internet Assigned Numbers Authority (IANA): system or well-known

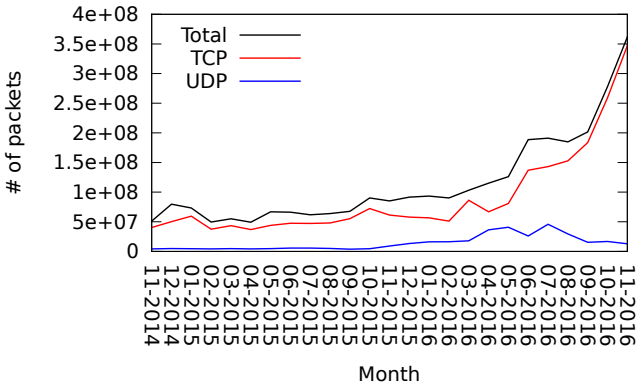


Fig. 1: Number of packets according to the months

ports (0-1023), user or registered ports/or vendors use for applications (1024-49151), and the dynamic and/or private or ephemeral ports (49152-65535). The difference uses of these ranges is described in RFC6335 [6]. Figure 2 shows the segmentation of the different scanned Port numbers during the 24-months. The system ports are frequently targeted compared to the user ports and dynamic ports. Indeed they contain the majority of well used standardized services, especially web services (HTTP and HTTPS) which are now a de facto standard for majority of applications. However, as attackers may also target specific and specialized ports, we conduct the study over all 2,125,080,102 TCP-SYN packets which represent 93,79% of TCP packets. Indeed, through priori analysis with the Suricata IDS[2] and our own tool [7], we discovered that most of TCP-SYN packets are related to port scanning even if other types of traffic like DDoS are also present. Discarding those few ones is risky as they are being mixed with scans. Keeping their related packets for analysis has zero impact since they are targeting a unique TCP port although our approach aims to find correlation over multiple TCP ports.

In this paper, we focus on the last four months from 08-2016 to 11-2016 which represent a higher number of TCP-SYN packets and more significant results. Table II represents the top-10 of probed ports in each month. We see that the services relating to traffic, servers and database are most targeted services in scans. However, these statistical results do provide the relationships between the probed ports.

Our aim is to discover, analyze the relationship groups of ports probed together, and so understand the attacker scanning strategies, especially by answering the following questions:

- Does it exist particular ports which are frequently probed in vertical scanning ?
- Does it target successive ports of the same segment or different ports of different segments ?
- Are there a strongly relationship between set of ports? Do they belong to the same targeted ecosystem services such as set of messaging or database services ?

[2]https://suricata-ids.org/

08-2016		09-2016		10-2016		11-2016	
Port	**Number**	**Port**	**Number**	**Port**	**Number**	**Port**	**Number**
telnet	106,891,746	telnet	122,793,736	telnet	183,601,219	telnet	229,054,687
ms-sql-s	7,788,167	3d-nfsd	9,615,076	3d-nfsd	19,775,823	pcanywherestat	28,614,678
ssh	3,921,299	microsoft-ds	8,285,083	microsoft-ds	7,975,819	3d-nfsd	26,786,260
ms-wbt-server	2,241,857	ms-sql-s	7,885,440	ms-sql-s	7,096,882	ms-sql-s	9,221,479
http	2,123,664	ssh	3,148,455	ssh	3,589,657	microsoft-ds	7,766,815
http-alt	1,257,436	ms-wbt-server	2,376,006	ms-wbt-server	2,479,072	ssh	5,290,784
https	1,238,662	http	1,729,630	http	2,019,860	cwmp	4,380,958
mysql	1,199,221	http-alt	1,052,039	https	1,121,846	http	3,262,717
dtserver-port	847,182	https	1,035,851	mysql	1,058,366	http-alt	1,111,872
smtp	464,722	mysql	950,356	http-alt	873,097	https	1,090,220

TABLE II: Top-10 of scan probes

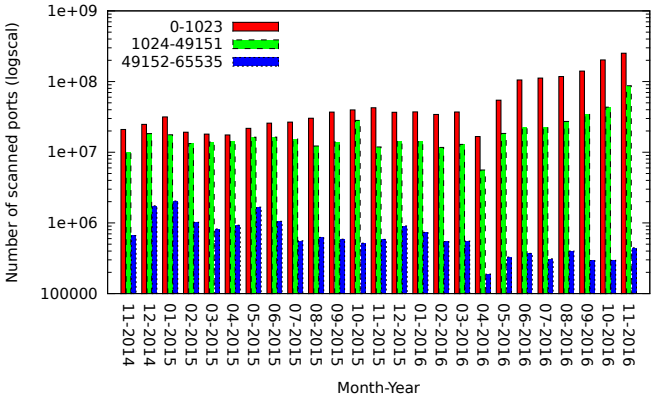


Fig. 2: Scanned Port numbers segmented on three ranges: System Ports (0-1023), User Ports (1024-49151), Dynamic and/or Private Ports (49152-65535)

In order to answer these questions, we propose a graph-based modelling approach for scanned ports through the time.

IV. Port-scan graph model

Different steps are needed for transforming TCP-SYN packets into graph describing TCP ports and their relationships from a scanning perspective. Source and destination IP addresses and ports have been widely used in the intrusion detection domain [13], [2]. They are good indicators of attack traffic flows and targeted services. In our context, we are interested in the scanning process to successive ports. Relying on a graph representation will allow to condense information and thus to model global knowledge from multiple instances, *i.e.* multiple scans involving different IP addresses.

A. Port numbers transformation and representation

Definition 1 (Port sequence): Let P be the set of all TCP ports. Assuming, a source and destination IP address, respectively IP_{src} and IP_{dst}, we denote $S((IP_{src}, IP_{dst}), T_s, T_e)$, the sequence of targeted ports by IP_{src} to IP_{dst} between the starting time T_s and the ending time T_e where $T_s < T_e$. S is thus a list of TCP destination TCP ports ordered by time: $S =< (p_1, t_1), \ldots, (p_n, t_n), \ldots >$, where $p_i \in P$, $T_s \geq t_i \leq T_e$ such that $t_{i+1} > t_i$.

Figure 3 shows an example of a port sequence with 5 targeted ports `{25, 26, 27, 443, 8080}` at time 50, 60, 70, 87 and 98 respectively. In our particular case of the darknet, no replies are sent back and no service are running. As a result, the sequences of ports thus only represents relations among attackers attempts rather than dependencies among multiple ports used by the applications [17].

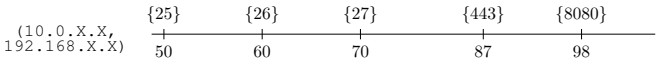


Fig. 3: Port sequence

B. Constructing a port-scan graph

In practice, the length of port sequences can be very long, and can reach up to a million of TCP port instances over one month. In addition, the ports can be redundant in a sequence in most of the time. To characterize causal relations between two successive ports in a port sequence, we introduce the notion of the *port-scan graph* model as an intuitive graph representation for successive scans in all port sequences.

A port-scan graph is constructed from multiple successive scans. In other way, it is a directed graph that represents successive relationships between targeted ports in port sequences. Specifically, each vertex represents a port number p_i and each edge (p_i, p_j) indicates that a scan after to port p_j occurs after a scan to the port p_i, where $p_i \neq p_j$. $p_i \neq p_j$ means that we neglect successive probes to identical part. As introduced in section III, the advantage of such an approach is automatically discard DDoS attacks from the analysis which are the main type of mixed traffic with scans in darknet data. Formerly, port-scan graph over a set of type of ports V is a labelled directed graph $G = (V, E, \beta)$:

- V is the set of observed port numbers from from P.
- E is a set of edges in G. Let p_u and p_v be be two port numbers in V. There is an edge $(p_u, p_v) \in E$ if and only if there exists a dependency rule $p_u \xrightarrow{l_{p_u,p_v}} p_v$
- β is a function that assigns for each edge (p_u, p_v) the number of dependency occurrence l_{p_u,p_v}.

The represented associations between all port numbers represent the signature features of all port sequences, i.e. the most and common behavior of IP source addresses targeting IP destination addresses.

V. PORT-SCAN GRAPH ANALYTICS

Graph algorithms or graph analytic are analytic tools used to assess structural properties of the graph and, as a final objective, to determine strength and direction of relationships between objects represented as vertex in a graph. For example, in a graph representing relationships (such as "interacting" or "probing" another port or IP address) between machines, graph analytic can to extract the following knowledge:

- the centrality of a node (TCP port), *i.e.* its predominant presence in shortest paths between other nodes (shortest scanning sequences)
- the clusters of ports being well connected in the graph, *i.e.* detecting dense partitions.

Such knowledge allows to answer the different questions raised in sectionIII.

To evaluate how relations among targeted ports evolved in time, our analysis is monthly-based. Hence, 24 unique graphs have been created from the two years of collected data. The built graphs contains of up to 6284 vertices and 589117 edges and very low density.

A. Centrality measures

Centrality score measure the communication importance of a vertex in terms of how central it is.

In this paper, we study two popular types of centrality: degree (baseline), and betweenness (flow-based).

1) Degree centrality: The degree centrality is the simplest and most popular centrality measure and gives a highly local view of the graph around each node.

Indeed it is the number of neighbors of a node. In a directed graph, each vertex has an indegree and an outdegree. Let $G = (V, E)$ and $\forall v \in V$. Indegree of vertex v is the number of edges which are coming into the vertex v. The indegree of v is denoted $deg^-(v)$. Outdegree of vertex v is the number of edges which are going out from the vertex v. The outdegree of v is denoted $deg^+(v)$.

2) Betweenness measure: The betweenness centrality measures are flow-related by considering the information flowing through edges. The most well-known centrality in this group is the Freeman's betweenness centrality [12]. It measures how much a given vertex lies in the weighted shortest paths of other vertices. Let $\delta_{st} = \delta_{ts}$ denote the number of shortest paths from $s \in V$ to $t \in V$, where by convention $\delta_{ss} = 1$. Let $\delta_{st}(v)$ denote the number of shortest paths from s to t that some $v \in V$ lies on.

$$betweenness = \sum_{s \neq v \neq t} \frac{\delta_{st}(v)}{\delta_{st}} \tag{1}$$

The betweenness centrality allows us to measure the probability that a sequence from the port s to the port t goes through the port v. s and t have the route of the ports sequences. Betweenness centrality is a measure of the influence or importance of a vertex over the flow of scanned ports between every pair scanned ports under the assumption that information primarily flows over the shortest path between them.

3) Results: We list top 10 port-scan patterns in Table III based on both centrality measures. The betweenness provides us the ports commonly used during the scanning, but not necessary most frequent ports, which thus differs our work from the usual study. We discover, for a months, there exists at least not-assigned targeted ports.

We discovered that the degree centrality and betweenness highlight some period of times where probes targets certain kind of services (close in their use):

- In 08-2016: the centrality concerns different services and protocols such as *http-alt: 8008* (HTTP Alternate), *irdmi: 8000* (Intel Remote Desktop Management Interface) or *memcache: 11211* (Memory cache service).
- In 09-2016: the centrality is around the computer network authentication protocols and distributed control systems such as *Kerberos: 88* and *d-s-n: 8086* (Distributed SCADA Networking), respectively.
- In 10-2016, the same as the last month, with additional printing protocol.
- In 11-2016, the centrality is around the transfer protocols such as *http: 80*, *https: 443*, *http-alt: 591*, *telnet: 23* and servers protocols.

There exists a large flow to network services, database service, computer network authentication protocols, and transfer protocol, in 08-2016, 09-2016, 10-2016, and 11-2016, respectively.

B. Community analytics measures

Community analytic algorithms aims to find dense subgraphs (also called clusters or communities) in a graph, in which its vertices are more connected within the cluster than with the nodes outside the cluster.

1) Modularity: Modularity measures the density of the partition of a graph into subgraphs called *modules*. It measures the density of links inside communities as compared to links between communities [15]. The modularity measures essentially compares the number of links inside a given module with the expected value for a randomized graph of the same size and same degree sequence.

Figure 4 shows the densely connected clusters of vertices, with sparser connections between clusters. The ability to detect such clusters of vertices is significantly important to discover the dense group of the common targeted ports at the same time whereas centrality measures are focused on individual ports. Groups within scanned ports graphs might correspond to consecutive probes communities, or service communities.

For computing modularity class, we use the algorithm developed in [4] for community structure discovery in large graphs.

The modularity of a partition of a graph [16] can be written as

$$Q = \sum_{m=1}^{M} [\frac{l_m}{L} - (\frac{d_m}{2L})^2] \tag{2}$$

where the sum is over the M modules of the partition, l_m is the number of links inside module m, L is the total number

08-2016		09-2016		10-2016		11-2016	
Betweenness	**Degree**	**Betweenness**	**Degree**	**Betweenness**	**Degree**	**Betweenness**	**Degree**
ms-wbt-server	http-alt	postgresql	ddi-tcp-1	unassigned	unassigned	distinct	distinct
http-alt	irdmi	xmpp-client	distinct	mit-ml-dev	mit-ml-dev	http-alt	ms-wbt-server
cbserver	pcsync-https	onscreen	unassigned	unassigned	ctf	ms-wbt-server	http-alt
unassigned	memcache	sdl-ets	kerberos	ctf	kerberos	cbserver	cbserver
irdmi	wap-wsp	ddi-tcp-1	mit-ml-dev	kerberos	xfer	quickbooksrds	dyna-access
puppet	sunproxyadmin	distinct	unassigned	pharos	unassigned	ncube-lm	http
fs-agent	ndmp	tram	d-s-n	unassigned	programmar	unassigned	quickbooksrds
sunproxyadmin	glrpc	ms-wbt-server	ctf	distinct	unassigned	unassigned	ncube-lm
trivnet1	unassigned	cluster-disc	xfer	xfer	npp	http	https
vcom-tunnel	tungsten-https	sip	sunproxyadmin	unassigned	distinct	dyna-access	telnet

TABLE III: Top-10 of services discovered with the betweenness and degree measures of port-scan graph

of links in the graph, and d_m is the total degree of the nodes in module m.

The principle of the algorithm is based on distinct community divisions process which is represented as follows:

1) Separate each vertex solely into m community.
2) Calculate the increase of Q for all possible community pairs.
3) Merge the greatest increase in Q.
4) Repeat 2 & 3 until the modularity Q reaches the maximal value.

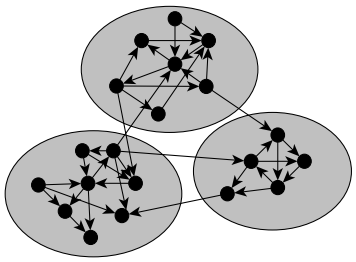


Fig. 4: Clusters of vertices

2) Results: Figure 5 represents the top-10 discovered clusters between 08/2016 and 11/2016. In each figure, the clusters are sorted in descending order of cluster size. The first line represents the number of targeted ports in different segments within the clusters. The second line represents the number of targeted ports which are assigned to specific services or not. Example: the port number `22` is assigned port number, which corresponds to the Secure Shell (SSH). In contrast, the port number `26` is an unassigned port number. In addition to extract strongly connected targeted ports, these graphs highlight the homogeneity or heterogeneity of clusters regarding the types of ports within a single cluster assuming one of two criteria, either IANA ranges or distinction between assigned and unassigned ports.

Clusters mainly contain user ports or vendors use for applications (1024-49151) but few of system ports (0-1023) and dynamic and/or private ports (49152-65535). It also exists clusters with only user/vendor ports (as for instance C6, C7, C8 and C10 in 5a). The same phenomena appears regarding the distinction between assigned and unassigned ports. As shown, there is no strong relationship between the homogeneity of clusters based on the two differentiation criteria. This supposes different underlying scanning strategies. In addition, through a manual inspection of clusters, we particularly discovered two types of strategy when targeted unassigned ports: either in a random or an incremental probing of TCP ports.

Furthermore, we analyze the heterogeneous and homogeneous clusters containing assigned ports. As clusters containing not-assigned ports, it also exists consecutive probes of services such as: user and private ports.

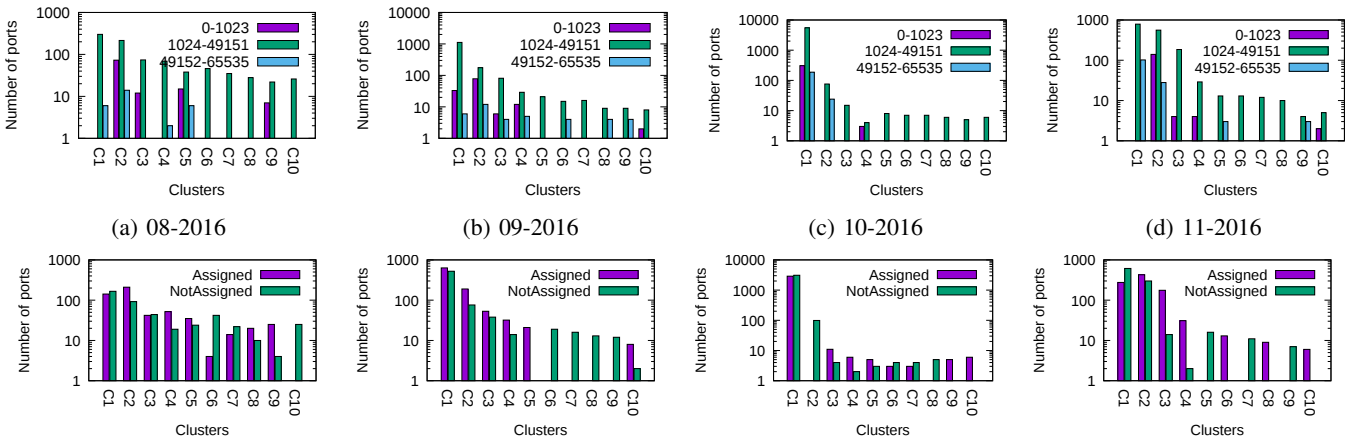
We discover other important clusters which contain non-consecutive probes and not randomly probes. These graph clusters are completely connected and contain particular scanned ports:

- A cluster containing both two types of services: database service ports and medical service ports. Database service ports such as *mysql: 3306*, *redis: 6379*, *ms-sql-s: 1443 (Microsoft-SQL-Server)*, *radg: 6789 (GSS-API for the Oracle)*, *ttc-ssl: 2484 (Oracle TTC SSL)*. Medical service ports such as *ohsc: 18186 (Occupational Health SC)*, and *biimenu: 18000 (Beckman Instruments, Inc)*. This cluster is shown in Figure 6.
- A cluster containing closest messaging transfer services such as: *103: Genesis Point-to-Point TransNet*, *20480: emWave Message Service*, *23: telnet*, and *2323: 3d-nfsd*. Genesis Point-to-Point TransNet: sometimes used with MS Exchange X.400 mail messaging traffic. 3d-nfsd is being used as an alternate telnet port.
- A cluster containing closest *Cisco systems* relative ports such as *gdp-port: 1997*, *x25-svc-port: 1998*, and *tcp-id-port: 1999*.

Those observation clearly indicates, first, that our method is able to find proper correlation (based on the semantics of the ports) and, second, that attackers may leverage intelligent scanning to attack a dedicated type of systems.

VI. Conclusion and future work

In this paper, we have proposed a new approach based on graph analytic techniques for grouping the scanned ports. We provide a method to help the security analysts to understand what clusters of services are commonly being targeted, the importance and semantic of clusters over the scanned ports, and different strategies of the port scanning. Our experimental results, over real data collected in a darknet, highlight the



(a) 08-2016 (b) 09-2016 (c) 10-2016 (d) 11-2016

(e) 08-2016 (f) 09-2016 (g) 10-2016 (h) 11-2016

Fig. 5: Top-10 clusters between 08-2016 and 11-2016

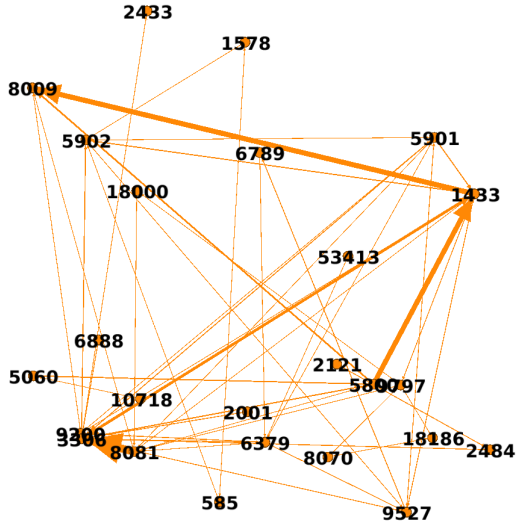


Fig. 6: Dense cluster of scanned ports.

ability of our method to discover unknown specific co-targeted ports belonging to the same or different types of services. Our future plan consists on performing a semantic distance measure over TCP ports and predict the future scanned ports.

Acknowledgments

This work was partially funded by HuMa, a project funded by Bpifrance and Region Lorraine under the FUI 19 framework. It is also supported by the High Security Lab hosted at Inria Nancy Grand Est (http://www.lhs.loria.fr).

References

[1] E. Balkanli and A. N. Zincir-Heywood. On the analysis of backscatter traffic. In *39th Annual IEEE Conference on Local Computer Networks Workshops*, pages 671–678, 2014.

[2] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani. Towards effective feature selection in machine learning-based botnet detection approaches. In *IEEE Conference on Communications and Network Security, CNS 2014, San Francisco, CA, USA, October 29-31, 2014*, pages 247–255, 2014.

[3] S. M. Bellovin. There be dragons. In *In Proc. UNIX Security Symposium III*, pages 1–16, 1992.

[4] V. D. Blondel, J. loup Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics Theory and Experiment*, 2008.

[5] R. Brewer. Advanced persistent threats: minimising the damage. *Network Security*, 2014(4):5 – 9, 2014.

[6] M. S. Cotton, L. Eggert, D. J. D. Touch, M. Westerlund, and S. Cheshire. Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. RFC 6335, 2011.

[7] M. Coudriau, A. Lahmadi, and J. Francois. Topological Analysis and Visualisation of Network Monitoring Data: Darknet case study. In *8th IEEE International Workshop on Information Forensics and Security - WIFS 2016*, Information Forensics and Security. IEEE, 2016.

[8] A. Dainotti, A. King, k. Claffy, F. Papale, and A. Pescapè. Analysis of a "/0" stealth scan from a botnet. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 1–14, 2012.

[9] Z. Durumeric, M. Bailey, and J. A. Halderman. An internet-wide view of internet-wide scanning. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, pages 65–78, 2014.

[10] C. Fachkha and M. Debbabi. Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization. *IEEE Communications Surveys Tutorials*, 18(2):1197–1227, 2016.

[11] J. Francois, R. State, and O. Festor. Activity monitoring for large honeynets and network telescopes. In *in "The Second International Conference on Internet Monitoring and Protection*, 2009.

[12] L. C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215 – 239, 1978.

[13] H. Hang, X. Wei, M. Faloutsos, and T. Eliassi-Rad. Entelecheia: Detecting P2P botnets in their waiting stage. In *IFIP Networking Conference, 2013, Brooklyn, New York, USA, 22-24 May, 2013*, pages 1–9, 2013.

[14] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage. Inferring internet denial-of-service activity. *ACM Trans. Comput. Syst.*, pages 115–139, 2006.

[15] M. E. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci USA*, pages 8577–8582, 2006.

[16] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004.

[17] B. Peddycord, III, P. Ning, and S. Jajodia. On the accurate identification of network service dependencies in distributed systems. In *International Conference on Large Installation System Administration: Strategies, Tools, and Techniques (LISA)*. USENIX, 2012.

[18] J.-P. van Riel and B. Irwin. Inetvis, a visual tool for network telescope traffic analysis. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '06, pages 85–89, 2006.