Machine Learning and Data Mining for IPv6 Network Defence

Michael Weisman[1], P. Ritchey[1], G. Shearer[1], E. Colbert[1], E. Dauber[2], L. Knachel[1], D. Sullivan[1], T. Parker[1] and R. Greenstadt[2]
[1]US Army Research Laboratory, Adelphi, USA
[2]Drexel University, Philadelphia, USA
Michael.J.Weisman2.CIV@MAIL.MIL

Abstract: In future battles, the warfighter will of necessity require more and more networked devices to perform a broad range of tasks. It has been predicted that by the year 2020, there will be 20 billion Internet-of-Things (IoT) devices (and more than 6.2 billion today) (N. Dragoni, 2017). IPv4 addresses are 32 bit and IPv6 addresses are 128 bit. All of the 232 ≈4.3 billion IPv4 addresses have already been exhausted, and except for the possible transfer from one device to another, and with the end-to-end design paradigm of IPv6, all new IoT devices will need an IPv6 address. Because of the huge number of potential IPv6 addresses (2128 ≈ 3.4 ×1038), probing every address is not possible. The only way determine IPv6 addresses is by watching traffic. In this paper, we will apply data mining and machine learning techniques to better understand the challenges of IPv6 security. We perform semi-supervised learning techniques such as augmenting k-means clustering with sparse labels to understand the distribution of IPv4 addresses, and explore whether or not clustering of IPv6 addresses is possible. We also will measure the performance of IPv4 anomaly detection algorithms and look to apply these algorithms with modifications to IPv6 data. Finally, we explore domain adaptation and transfer learning from IPv4 to IPv6 and ask how easily can we adapt a system trained for IPv4 to IPv6 and what changes do we need to make? If we include additional IPv6 training data, how do things change?

Keywords: data mining, machine learning, IPv6, Internet of Things

1. Introduction

Work on understanding IPv6 security has already begun, but compared to what we know about IPv4 attacks and defences, the IPv6 landscape remains a mostly uncharted area. In this paper, we give a brief comparison of the IPv4 and IPv6 protocols, and propose applying unsupervised learning and semi-supervised learning algorithms to discover the current state of IPv6 topologies. We discuss recent work in visualization, anomaly detection, and intrusion detection system modelling within the Network Sciences Division at the US Army Research Laboratory. We also give a short survey of unsupervised machine learning, and suggest how these algorithms can be applied to defending against IPv6 attacks.

2. Army relevance

The military is increasingly relying on computers and networks for its operations, and security is of utmost concern. The warfighter of the future will likely be equipped with more than one or two internet devices, and the internet of battlefield things is quickly becoming a reality. Because the supply of IPv4 addresses will soon be exhausted (except possibly a few that get recycled) the warfighter of tomorrow will likely need to use IPv6 for global communications.

Cyber security research at the US Army Research Laboratory (ARL) must always be relevant to the needs of the Army. Theatres where the Army operate are diverse and complex (Kott 2016), made up of allied and civilian personnel and assets as well as adversary forces. Allied assets are subject to capture, and must be secure and resilient. In addition to fundamental research, the Army Research Laboratory has also been involved in research for operations, providing expertise to numerous organizations (Kott 2016).

3. IPv6 vs IPv4

The introduction of IPv6 was designed to improve upon IPv4 in four key areas as well as bring additional security measures. These four areas are (Deering and Hinden 2017)

- **Expanded addressing capabilities:** With IPv6 the IP address size is increased from 32 bits to 128 bits resulting in potentially $3.4 \cdot 10^{38}$ addresses.
- **Header format simplifications:** The IP header and the extension headers each have a ``next header'' field (Kozierok 2005) so that headers may be chained together, each header pointing to the next and the last header will indicate what the next upper level protocol is.

- **Improved support for extensions and options:** By dropping some of the IPv4 header fields, making others optional, and by introducing optional headers, IPv6 allows for greater flexibility and permits more efficient forwarding. Traffic flows can now be labelled so that a sender may make special demands on delivery based on these labels.
- **Flow labelling capability:** Now packets may be labelled by membership of traffic flows and where a sender requests special handling or when a sender requests a particular quality of service.

IPv4 and IPv6 perform the same basic task. The mechanisms used to transport packets across the network are mostly unchanged, and the protocols at higher layers that transport application data are minimally affected by IPv6. The same logic can be applied to threats and vulnerabilities at higher layers. Application vulnerabilities and exploits will not typically be affected by the use of IPv6.

Many modern systems are dual stack (Holder 2017). This means the existing vulnerability surface already includes the IPv6 space, even for networks that are nominally IPv4 only.

Vulnerabilities that are similar between IPv4 and IPv6 include application attacks that occur at the application layer. Other attacks that continue to exist include name resolution attacks and flooding. The potential for Man-in-the-Middle attacks still exists, in some familiar and new forms, all of the familiar routing threats (hijacking, denial-of-service, etc.) still exist. Finally, fragmentation attacks to evade Intrusion Detection Systems or application firewalls are still a real threat, as fragmentation has no effect on network firewalls.

Below we list some of the new security challenges between IPv4 and IPv6 and provide some brief comments.

- **Address Reputation** techniques must be revised. Hierarchical reputation based on subnet ownership may still be viable, within limits.
- **Internet Control Message Protocol version 6 (ICMPv6)** functions differently in IPv6. ICMPv6 packets are used for path discovery, multicast group management, and neighbour discovery, in addition to error and diagnostic messaging. Therefore, IPv6 needs ICMPv6 packets to function. IPv4 networks sometimes drop incoming ICMP packets as a potential security threat.
- **Neighbour Discovery Protocol (NDP) and Dynamic Host Configuration Protocol version 6 (DHCPv6**) create new threats primarily for router hijacking, denial of service. The threat can be controlled within a local network behind a firewall. A mobile network is more challenging however.
- **Extension Header** layout is new to IPv6. Programs must parse the IPv6 header in a different manner than the IPv4 header. The header contains a chain of variable length, which is difficult to inspect.
- **Profiling/Reconnaissance** of IPv6 networks must rely more heavily on DNS or NDP/DHCPv6 if available.
- **Changes to privacy and transparency** including the use of privacy addresses, opaque static addresses, and cryptographically generated addresses.
- **Firewall tuning and maintenance** is necessary without the presence of NATs. IPv6 is designed with end-to-end connectivity and global addresses everywhere, therefore there is no longer a need for address translation.

With the introduction of IPv6 came the introduction of optional extension headers. The *Routing Header Type 0* contains a list of addresses that the traffic must traverse on the way to the receiver. One clever attack is to list a pair of addresses multiple times causing increased traffic on the segment between these nodes as the packet ping pongs back and forth between them. This could overwhelm one or both of the nodes and cause a denial of service attack. Because of the potential for this type of attack, the routing extension header was ultimately discontinued (Ullrich et al. 2015, Abley et al. 2017).

The IPv6 fragmentation scheme, specified in RFC 2460 (Deering and Hinden 2017) allows IPv6 packets to be broken up into smaller parts and then reassembled at their destination (Gont 2017). These fragments can overlap and thus cause ambiguities in the packet reconstruction. Attackers could exploit this to evade Network Intrusion Detection Systems (Gont 2017). IPv6 is potentially susceptible to attacks on two types of fragmentation. One type, overlapping fragmentation is now forbidden in IPv6. Atomic fragments, packets used in IPv4 containing one fragment used in protocol translation can cause dropping of benign fragments in IPv6 with the same identifier (Ullrich et al. 2015).

Ostensibly, one could come to the conclusion that the huge address space of IPv6 could protect networks from attacks that choose random addresses. The need for administrators to be able to track addresses of sub-networks and record accurate logs necessitates manual address assignments for servers and routers (Ullrich et al. 2015). Because IPv6 requires router and neighbour advertisements *a priori*, securing ICMPv6 via IPv6 has so far been lacking and protection must be provided on other layers (Ullrich et al. 2015).

Reconnaissance in IPv6 is much more challenging than in IPv4. Previous methods such as eavesdropping or brute force search presently do not seem viable do to the huge address space in IPv6. New methods for active probing will need to be developed and data sets of IPv6 addresses will need to be collected (Ullrich et al. 2015).

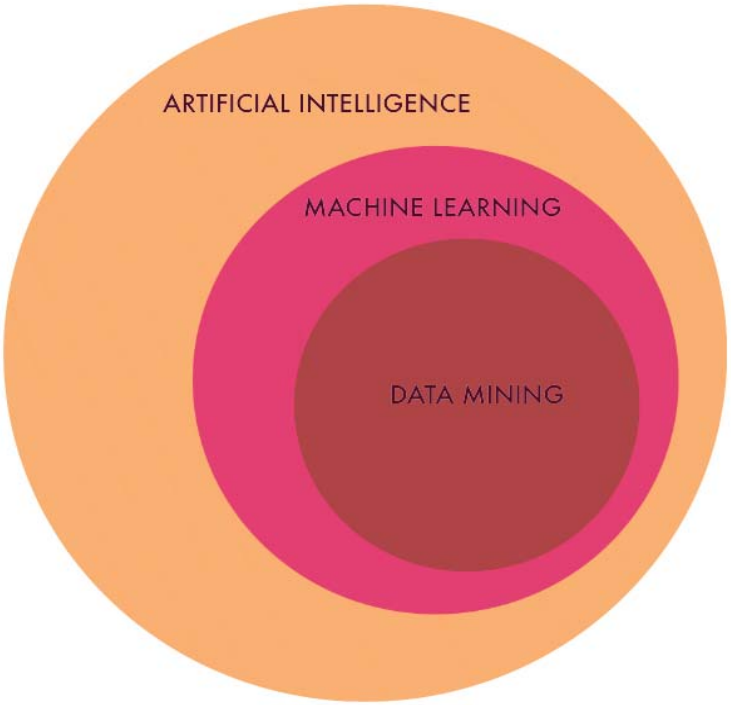

Figure 1: Artificial intelligence, machine learning, and data mining

4. Artificial intelligence, machine learning, and data mining

As illustrated in Figure 1, machine learning can be thought of as a subfield within the larger field of Artificial Intelligence (AI) and similarly, it can be argued that data mining is a subfield within machine learning. AI is concerned with taking machines (for example robots, computers, smart phones) and giving them intelligence: the ability to learn and reason as a human or a super-human. For example, a computer can learn to play chess better than the level of a grand master or teach itself to play go better than its predecessor that handily beat the top player!

Perhaps one of the most famous tests of AI is the "Turing Test" which has been described by Turing as the following (Turing 1950): Put a human and a computer behind a wall, give each a typewriter or a printer, and ask each a set of questions. If an "average" person cannot tell the difference between the human and the computer, the computer has passed the test.

Modern mainstream AI researchers have tended to downplay the importance of such a test for a number of reasons. Among them are:

- People are easily fooled. It is not so difficult in this scenario to cause a person to believe a computer is human or a human is a computer.
- There are many more important applications of AI then convincing someone that a computer is a person, such as: automatic object or character recognition, solving scheduling and resource allocation problems, and automatic intrusion detection!

Early on AI met with some failures, perhaps due to the limits in computer power and memory. Recently there has been an upsurge in interest in AI do to a number of successes in the area of translation and automation.

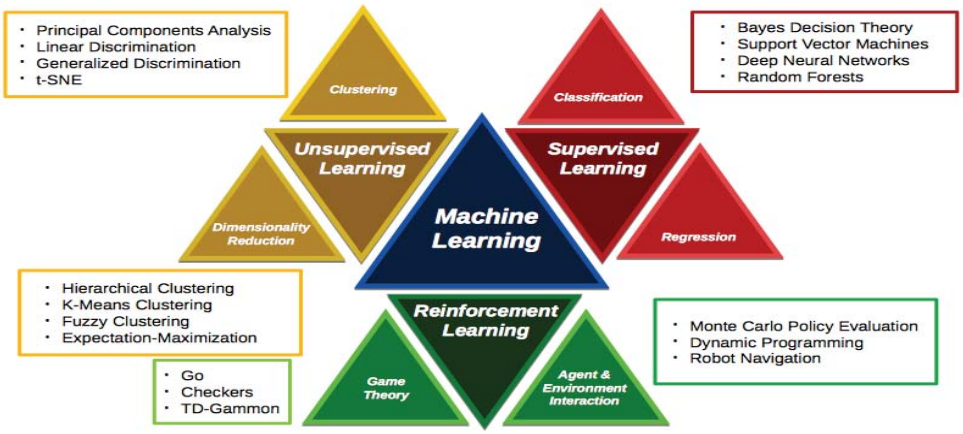

Figure 2: Machine learning categories inspired by LinkedIn [17]

The three main categories of machine learning are supervised, unsupervised, and reinforcement learning. Data mining is concerned with the analysis of large amounts of data, and searching for patterns. Supervised learning and unsupervised learning each aim to mine data to either predict new occurrences of data, or find subsets or other patterns within the data. Supervised learning occurs when we are given data as well as labels that indicate to which class the data belong. Examples of supervised learning are classification and regression. Classification is the process of inferring the class membership of new data, based on what has been learned about the class membership of existing data. Regression is the process of fitting lines, curves, or higher-dimensional surfaces to data. A model is specified (e.g. linear or exponential) and the coefficients defining the curve or surface are estimated. Unsupervised learning is the process of categorizing unlabelled data. Clustering algorithms attempt to group data together that belong to a common class or are close in feature space. Dimensionality reduction is an attempt to explain high dimensional data by reducing the defining features to a lower dimensional space. Of the three categories, reinforcement learning most exemplifies feedback-based learning. An agent takes a series of actions, and is rewarded for actions that take it closer to its intended goal. An example of reinforcement learning is the training of a robot to find its way through a labyrinth or cave, rewarding it for decisions at each step along the way. Dynamic programming is applicable here, mathematically reducing an optimization problem to a nested series of sub-problems. Given a reinforcement learning task for multiple agents, each agent chooses its optimal rule based on the anticipated actions of the other agents. The procedure can often be formulated as a multiplayer game. Each agent has its objectives defined to support the overall mission.

The ***k-means algorithm*** forms clusters by starting with an initial set of cluster centroids (means) and forming cluster memberships by minimizing the distance from each data point to the mean of the cluster it will belong to. The means are then updated and the membership set is recomputed. The algorithm continues iterating until convergence. The ***fuzzy k-means algorithm*** is similar, with the additional property that a data point could belong to more than one cluster via *fuzzy* membership. Each point's membership in each set is computed such that the total membership among all sets adds to one. The *fuzzy k-means algorithm* has an additional parameter b that determines the *fuzziness* of the memberships. A value b=1 corresponds to traditional k-means with no fuzziness and higher values of b allow for more fuzziness. Shown in Figure 3 are solutions for values of b ranging from $\sqrt{2}$ to 8. The data shown is just a notional, but the clusters could represent benign and malicious data, for example. The white circles represent the *fuzzy* cluster means.

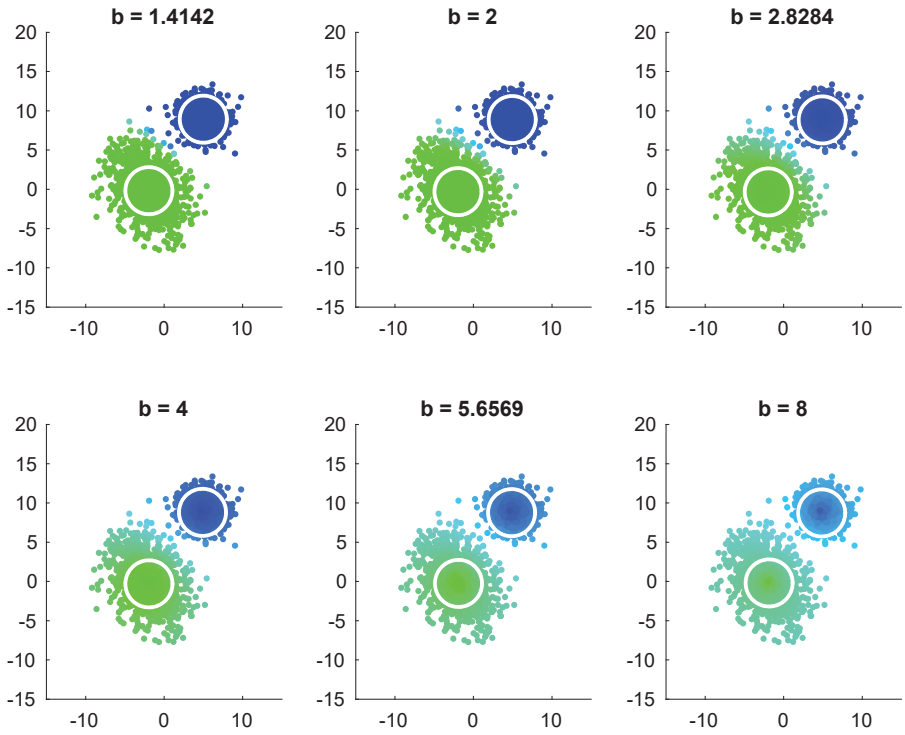

Figure 3: ***Fuzzy k-means clustering***. The parameter b ranges from √2 to 8 and controls the *fuzziness* of the cluster set memberships

Transfer learning is an area of machine learning concerned with applying knowledge gained from one problem to solve another related problem (Pratt 1993). Specifically, we are interested in transferring knowledge gained from IPv4 related problems to solve IPv6 related problems. If we are successful, we can use IPv4 tools and algorithms to build new versions to address IPv6 problems. Domain adaptation is an area of machine learning concerned with using data from one source to learn about data from another related source (Daume 2006). Specifically, we are interested in training on existing IPv4 datasets to learn to solve related IPv6 problems. If we are successful, we can train our tools on already collected and studied IPv4 data, decreasing the data collection burden for our new research. These two areas of research are highly related, as both exploit the relationship between the two sources of data. As a result, the ability to apply techniques from either research area to IPv6 is dependent on the level of similarity between data from IPv4 and data from IPv6 with respect to the features necessary to solve a given problem.

5. Examples of ARL network sciences division tools

We briefly discuss two of the many tools developed and used by analysts at ARL for intrusion detection and also describe some of the visualization work we do.

Interrogator is an intrusion detection system (IDS) developed by ARL as both a highly scalable, flexible capability to support their Cybersecurity Service Provider (CSSP) environment and ongoing research (Kott 2016). The architecture behind **Interrogator** facilitates easy implementation of new tools, technology and ideas that allow the system to adapt as adversarial tools and techniques evolve. The data collection techniques, which are unique to **Interrogator**, facilitate studies by providing the types of information and data required to perform sound, scientific research. For CSSP use **Interrogator** provides a user interface that allows analysts to view, query and deeply dive into collected information prior to filing incident reports using the reporting workflow capability. The centralized nature of the architecture benefits both analysts and researchers by providing the data that is needed in the quantity suitable for the intended purposes. Although created before IPv6 was widely used, **Interrogator** was designed from the ground up to support IPv6 network addresses from data capture, processing through final storage and data display to the analysts. A diagram depicting basic **Interrogator** functionality is shown in Figure 4.

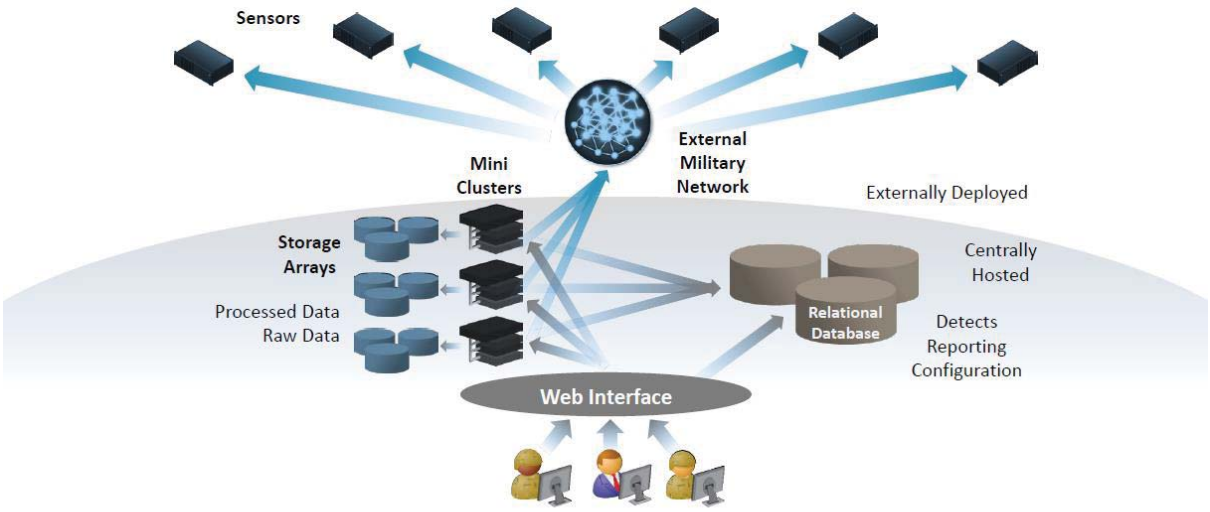

Figure 4: Notional diagram of the ARL interrogator system

Logalyzer is a Python framework developed by ARL providing a plug-in capability for quickly evaluating different machine learning algorithms with different types of input files. By treating each step (data processing, machine learning and results generation) as plug-ins, the researcher can instantaneously switch between plugins at the command line when **Logalyzer** is executed. Internally, **Logalyzer** utilizes an API that was developed to standardize how data is internally transferred between each step, allowing researchers or developers to quickly add new plugins supporting their requirements if one doesn't already exist. Rather than explicitly categorizing IP addresses, **Logalyzer** treats addresses simply as strings. **Logalyzer** is flexible enough though, that depending on the machine learning and data mining algorithm being implemented, logic for discriminating between IPv4 and IPv6 addresses can be handled.

Visualization is used by security analysts to assess trends and patterns in large volumes of network traffic information. As such, visualization is a key tool to building an ability to understand the network threat environment. Figure 5 depicts visualizations of a few standard network traffic interactions. For example, in the upper right hand corner (labeled HTTPS Interaction) we see one central node that represents a web server interacting bidirectionally with clients accessing the website and receiving information back. The mail servers (middle right) are the two central nodes that have multiple nodes (mail clients) communicating with them. The heavily weighted arrow between the two mail servers shows the direction of the majority of mail transmitted. The DNS Server depicted in the lower left shows the central node receiving DNS lookup requests and sending back DNS lookup results. Although IPv6 traffic is already present on many networks, incorporating IPv6 into exiting IPv4 visualizations can be challenging, from both a technical and practical aspect. Existing software for IPv4 network visualization may not adequately handle IPv6 . Whereas the 32 bit IPv4 addresses could generally be converted into an integer for mapping with relative ease, the 128 bit IPv6 address can be more difficult to work with if straightforward integer mapping is attempted by the visualization software. Additionally, IP header fields have changed between IPv4 and IPv6, potentially complicating processing. Setting aside software implementation issues, many existing visualizations of IPv4 space as a whole, such as mapped Hilbert curve, cannot be directly converted to IPv6 space. Furthermore, certain assumptions made about IPv4 space tend to be broken by IPv6. One is the degree of sparsity. The IPv6 space as a whole is extremely sparse. At present only 1/8 of the overall address space has been allocated for use (Ripe 2009). Within this allocated space, blocks containing 2^{64} addresses (compared to 2^{32} unique address in the entirety of IPv4 space) are assigned at the lowest level, to customer sites or individual organization networks (Ripe 2009). Thus, IPv6 space is extremely sparse. In Figure 5, IPv6 addresses are represented as nodes, and the packet flows between them are represented as edges. This flow graph is based on an IPv6 traffic sample, capturing some common network behaviour. A flow graph is ideally suited to visualize interactions between a set of hosts where communication may be web-like and nonlinear.

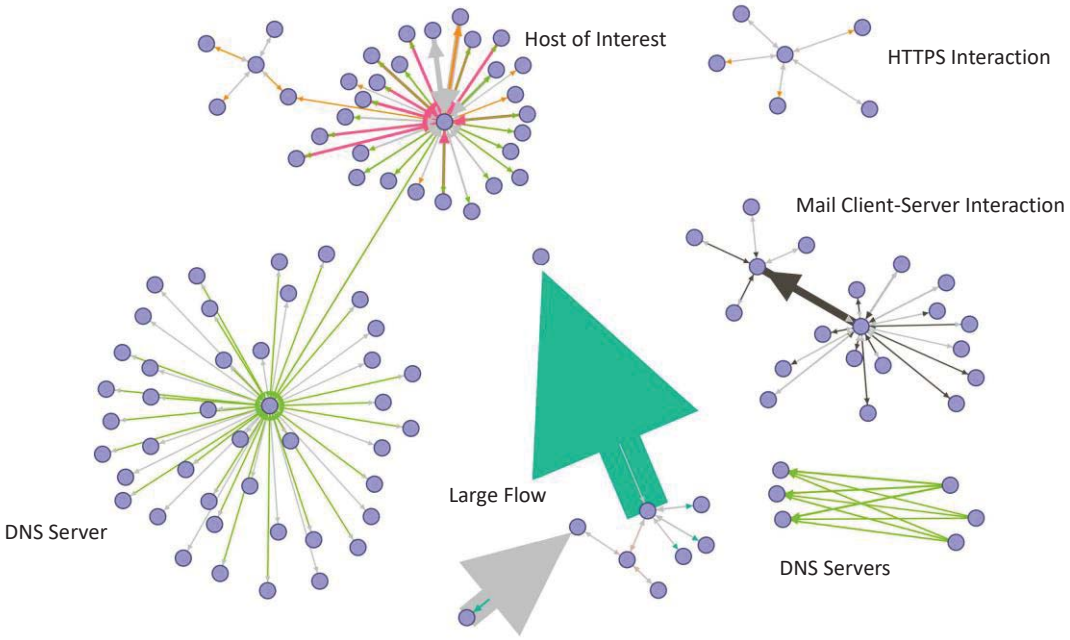

Figure 5: An IPv6 network constructed from real data

Despite the challenges presented by IPv6, most network visualization methodologies should still be fundamentally applicable. For network visualizations constrained to a local network or a relatively small number of unique addresses, existing flow mapping and network diagramming techniques such as node-edge graphs should be sufficient. For applications which seek to model IPv6 space at a larger scope, alternative visualization techniques have been proposed, such as whitespace filtering (sparsity reduction, in essence) to compensate for the sparsity of the IPv6 address space and hierarchical tree maps (Barrera 2009, Nakamae et al. 1999) to more clearly define the association of a given IPv6 address by leveraging IPv6 allocation standards.

6. Future work: Applying algorithms to IPV6 security

We have previously discussed some similarities and differences between IPv4 and IPv6, but in order to apply transfer learning or domain adaptation techniques to a specific problem we need to identify the specific IPv4 features used for that problem and determine if the same information is still available in IPv6. If it is available in some form, it may be possible to apply some combination of transfer learning and domain adaptation to solve that problem. If not, then we will likely need to do additional research to find an alternative solution specific to IPv6. A metric known as distortion can be used to measure the difference between two datasets to determine if application domain adaptation techniques are necessary (overdorf2016blogs). We may be able to use or adapt this metric to determine if application of domain adaptation or transfer learning techniques is possible for a given problem based on the associated IPv4 features and comparable features for IPv6.

Our approach is to measure the performance of IPv4 anomaly detection algorithms using machine learning techniques. We will establish a repository of labelled IPv4 data for each type of anomalous traffic and cyber attack (e.g. Distributed Denial of Service, flooding, and exfiltration). For detection algorithms with a high success rate, we will determine the connection between the features and the resulting detection outcome. We will then stand up a testbed and re-create the malicious network traffic using IPv6 data. For realism, the testbed will run emulation software to operate the firmware of production network elements. From the captured data of the malicious IPv6 traffic, we will assess the accuracy of the machine learning detection algorithms. In addition to comparing how the detection algorithms classify the IPv4 and IPv6 traffic, we will note whether or not the anomalous traffic behaves differently in the IPv6 infrastructure. Based on the results of comparing detection algorithms applied to each cyber attack in an IPv4 and IPv6 network, we will identity the features which have the broadest scope and accuracy. In addition, we will test how security appliances can be configured to enable IPv6 specific features (e.g. ICMPv6) while still protecting their domains. This information will benefit information assurance analysts to prepare to instrument and defend their networks when they transition to IPv6. Finally, we will explore domain adaptation and transfer learning from IPv4 to IPv6 and ask how easily can we adapt a system trained for IPv4 to IPv6 and what changes do we need to make? If we include additional IPv6 training data, how

do things change? If we can use features that have been shown effective in the domain of IPv4 for IPv6 data, or do we need to transform the features in some way?

References

Abley J., Savola P. and Neville-Neil, G. (2017) "Deprecation of Type 0 Routing Headers in IPv6," **RFC 5095** (Proposed Standard), IETF.

Barker, K. (2017) "The Security Implications of IPv6," http://www.sciencedirect.com/journal/network-security/vol/2013/issue/6, Accessed: October 2017.

Barrera, D. (2009) "PC Security Visualization Tools and IPv6 Addresses," *International Workshop on Visualization for Cyber Security*.

Daume III, H. and Marcu, D. (2006) "Domain Adaptation for Statistical Classifiers," *Journal of Artificial Intelligence Research* 26, pp. 101-126.

Deering S. and Hinden, R. (2017) "Internet Protocol, Version 6 (IPv6) Specification," http://www.rfc-base.org/txt/rfc-2460.txt, Accessed: October 2017.

Dragoni, A. G. (2017) "The Internet of Hackable Things," in *arXiv:1707.08380v1,* Accessed: October 2017.

Gont F. (2017) "Processing of IPv6 Atomic Fragments," http://tools.left.org/rfc/rfc6946.txt, Accessed: October 2017.

Hagen, S. (2014) **IPv6 Essentials**, O'Reilly, 3rd edition.

Hinden, R. and Deering S. (1998) "IP Version 6 Addressing Architecture," http://www.rfc-base.org/txt/rfc-2373.txt.

Holder, D. (2017) "Ipv6 Security Fundamentals," **UK IPv6 Council Security Workshop**.

Kott, A. (2016) "Overview of Cyber Science and Technology Programs at the US Army Research Laboratory," **Journal of Cyber Security and Information Systems**, Vol. 5, No. 1.

Kott, A., Swami, A., and B. J. West (2017) "The Internet of Battlefield Things," **Computer Magazine, IEEE Computer Society.**

Kozierok, C. (2005) **The TCP/IP Guide**, No Starch Press.

LinkedIn (2017) "Business Intelligence and its Relationship with the Big Data, Data Analytics and Data Science," https://www.linkedin.com/pulse/business-intelligence-its-relationship-big-data-geekstyle, Accessed: October 2017.

Nakamae, S., Sekiya, Y. and Murai, J. (1999) "A Study into a Visualization of an IPv6 Network."

Overdorf R. and Greenstadt, (2016) "Blogs, Twitter Feeds, and Reddit Comments: Co-Domain Authorship Attribution," PoPETs 2016(3), pp. 155-171.

Pratt, L. Y. (1993) "Discriminability-based Transfer between Neural Networks," in *Advances in Neural Information Processing Systems*, pp. 204-211.

Ripe NCC (2017) "IPv6 Address Allocation and Assignment Policy," **Ripe-684**.

Turing, A. M. (1950) "Computing Machinery and Intelligence," Mind LIX (236), pp. 433-460.

Ullrich, J., Krombholz, K., Hobel, H., Dabrowski, A. and Weippl, E. (2014) "IPv6 Security: Attacks and Countermeasures in a Nutshell," 8th USENIX Workshop on Offensive Technologies.