# PEIMA: Harnessing Power Laws to Detect Malicious Activities from Denial of Service to Intrusion Detection Traffic Analysis and Beyond

Stefan Prandl
Curtin University

July 24, 2017

**Abstract**

Distributed denial of service attacks (DDoS) are a constant problem of network operators today. Thanks to low cost of entry, high effectiveness, and the difficulty present in filtering out such attacks from inbound network traffic, DDoS attacks are relatively common and difficult to mitigate against.

Recent discoveries regarding the conformity of network traffic to certain power law distributions, namely Benfords and Zipfs laws, has allowed us to develop a new method of denial of service detection based entirely on packet header inspection. Power law distributions are fascinating artefacts of natural processes, applications of which can be found in anywhere from word counts in books through to numbers used in bank statements. Our research can detect DDoS attacks by using such distributions to detect strongly unnatural network traffic scenarios with only minimal metadata. This however, is not the whole story. Power law potential in IDS is largely un-researched, and could be applied for more general anomaly based IDS purposes. It can even be used to filter for denial of service packets in live streams of data.

What makes Power Laws both fascinating and interesting is that they have an inbuilt resistance to attempts to tamper or subvert the data analysis. Given the low computational cost associated with Power law processing and the foolproof security inherent to the methods, Power law distributions make perfect tools for cyber defense, especially in the areas of DoS and intrusion detection. In this talk we will introduce and discuss the significance and power of power law distributions, how they relate to computers, and how this can be used to develop new anomaly detection systems.

## 1 Introduction

Network based attacks continue to be significant threats for organisations worldwide. Attacks like the WannaCry or Petra/GoldenEye ransomware worms,

along with the ongoing problem of stockpiled zero day exploits being stolen by unsavoury interests, indicate the continued requirement for intrusion detection systems capable of detecting previously unseen attacks [7]. Furthermore, attacks like the KrebsOnSecurity DDoS amplification attack demonstrate the rapidly growing problem that the internet of things presents [1]. First line defences such as firewalls and signature based defenses are usually ineffective against these kinds of attacks, usually because they are targeting services that are being hosted by the victim organisation, or because the user is the primary vector as is the case in social engineering scenarios (spearphishing, etc) [12]. As such, anomaly based detection systems based on determining the difference between usual "benign" behaviour and malicious behaviour generally through application of some kind of learning based agent, are becoming more desirable.

Anomaly based detection systems have problems though. They require training to determine what is and is not malicious, they require constant tuning as systems and services change and evolve, and they produce many more false positives than signature based systems. Very often they also require specialised hardware to collate and correlate activities across an entire network. These problems tend to exist because anomaly based detection systems need to learn how a network works. It seems obvious that each system will behave differently, as each system is uniquely constructed, and has differing services and users. As such, anomaly based detection systems must learn and continue to learn what the normal state of any given system is such that it can determine what an unusual state is. This predisposition for false positives and requirement for ongoing learning creates a high maintenance environment that businesses are less than excited in maintaining, given that false positives and training time both use resources and desensitise a business to alerts [11].

An alternative way of setting up an anomaly based detection system is against some known law or known behaviour of a system. This white paper discusses a new such set of laws that can be applied, specifically toward network traffic. These are power law probability distributions, which are related to natural processes, and can be used to determine the difference between benign traffic and classes of malicious traffic without any training. This white paper will introduce power law probability distributions, discuss how to properly apply and determine fitness to them, and then discuss potential applications and how such a system could be set up.

# 2 Power Law Probability Distributions

## 2.1 An Introduction

Power law probability distributions are a class of probability distributions that follow the general exponential decay profile of $n^{\frac{1}{x}}$. The interesting and useful aspect of power law probability distributions is that they tend to be found in

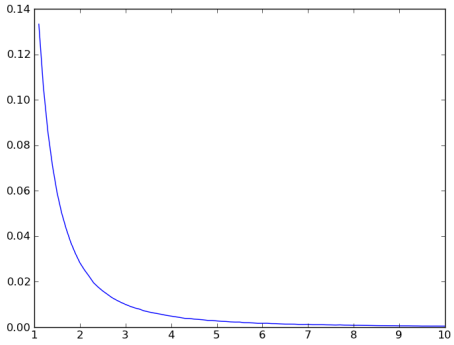


Figure 1: A visualisation of the Pareto Distribution

functions descriptive of, or dependant on a natural process [9]. They also happen to be very sensitive to changes to that natural process, very rapidly distorting away from the expected distribution as the process is distorted. It is rather hard to avoid this distortion, as both humans and computers are generally bad at approximating natural randomness. This makes power law probability distributions highly effective in determining when a natural process has been interfered with, such as in the case of fraud [3].

While fraud detection use of power laws in financial environments is fairly well researched, power law applications to information security is reasonably new. This is because computers were not thought to have natural processes, being entirely artificial constructs. However, natural processes have no requirement to be derived from nature; while it can be argued that economics is an extension of human nature, financial transactions do not appear in nature. At the very least it has been determined that, collectively, network traffic is likely to approximate a natural process. It is possible that other elements of computer function also approximate natural processes, however these have not been researched yet. Naturally, this white paper cannot authoritatively discuss all possible applications of power law probability distributions to security. What this paper can and will discuss is how to build a bespoke system to compliment existing network solutions.

## 2.2 Building Detectors

There are two kinds of probability distribution generally. Continuous distributions, which are interested in the general distribution of real numbers; and discrete distributions, which are interested in the distribution of quantities. In order to use power law probability distributions for detection purposes, one must

determine how well one's data fits the distribution. Statistically, it is far easier to determine such fit with discrete distributions than continuous. So while it is possible to build a power law detection device that relies on continuous power law probability distributions (like for example, the Pareto distribution as displayed in Figure 1), such construction will be left as an exercise for the reader.

To determine how well data fits any given distribution, one must use a goodness of fit function. Such a function for a discrete distribution can be as simple as determining the difference between what is seen and what is expected to be seen. The simplest function is the sum of squares function, which adds together the squares of the differences between what is seen and what is expected for each element of the distribution, like so:

$$SSE = \sum_{i=1}^{n} (observed_i - expected_i)^2 \tag{1}$$

This is an inherently simple calculation, and is not the most sensitive, however it is still functional for detective purposes. It has been suggested that the most sensitive detection method for power law probability distributions is the Watson's Cramer von Mises test for goodness of fit (details on how to implement can be found in [6]), which is a far more complicated method. If possible, it is recommended that external libraries are used to perform these kinds of assessments should they be desired.

In order to determine when an anomaly has occurred, a threshold under which deviation from the power law is accepted and above which an anomaly is reported must be set. In our research [10] a numeric threshold value of 0.163 for the Watson's goodness of fit test is considered to be most effective for denial of service (DoS) detection, however there is no reason to believe this is the best possible value, and other thresholds as low as 0.01 have been suggested [2]. These relate to probability thresholds of 0.01 and 0.5 respectively. Experimentation is advised, especially as some goodness of fit functions that can be used produce wildly different statistical values.

As this is a statistical value being used for detection purposes, a catchment window size must be set. This can be of any given size, however it should be noted that the larger the window, the less sensitive to short events a detector will be. Conversely, the smaller a window, the more sensitive to short events a detector will be. The size of a window also can increase the number of false positives detected, namely that small windows can be so sensitive that short changes in traffic appear as attacks. The recommended window size is approximately 1000 packets in total.

## 2.3 Zipf's Law

In its more general form [8], Zipf's law states that if elements are ranked by frequency, then the probability of occurrence of an element of rank $r$ should be

approximately determinable by the following:

$$P(r) = \frac{1}{r^{\beta}} \times P(1), \tag{2}$$

For our purposes, we consider $\beta$ to be equal to 1. Zipf's law is best used with groups of items. A basic network case would be "number of packets of this type/size/flag/etc". Theoretically, the second most populous group should have approximately $\frac{1}{2}$ the number of items as the first. The third group should have $\frac{1}{3}$ the number of items as the first, and so on. This distribution is dependant on the population of the most populous group in each window, and will need to be recalculated every time. Obviously, this is going to be a reasonably slow part of this calculation, and is the reason that while this can be conceptually easier to apply to metrics it is less capable than Benford's Law.

### 2.4 Benford's Law

Benford's law [6] states that the probability of occurrence of each leading digit $d$ ($d \in 1...9$) of a large set of decimal numbers generated by a natural process is given as

$$P(d) = \log_{10}(1 + \frac{1}{d}). \tag{3}$$

Note that the obtained probability distribution is not dependent on the dataset being measured. Further, Benford's law may also be used to predict the distribution of first digits in other bases, e.g., binary, besides decimal, and also for the second and subsequent digits. The following gives the predicted frequencies for the first digits of decimal numbers.

| Digit | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Prob. | 0.301 | 0.176 | 0.125 | 0.097 | 0.079 | 0.067 | 0.058 | 0.051 | 0.046 |

Benford's law does not change with regard to the data it collects, the values above are the same for every benford compliant data set. This means one can hard code the expected quantities as a function of their catchment window size. This makes detecting with Benford's law mostly a bucketing exercise, and consequentially very fast. The drawback is that Benford's law is more difficult to apply to metrics, as it deals with numbers that are generated by natural processes. An example from prior research is that Benford's law is applicable to the time between arrivals of different network packets. If one finds a metric that conforms to Benford's law, this is usually of far better use in detective terms.

## 3 Applicability to Security

### 3.1 Overview

While the above can be applied to essentially any computing environment, assuming that at first you have determined that the metrics you wish to use

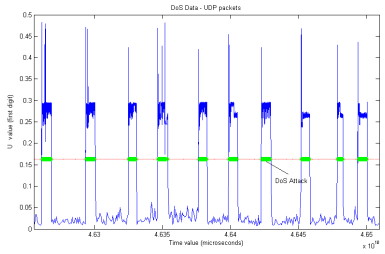


Figure 2: Benford's law analysis of packet inter-arrival times during several short DoS attacks

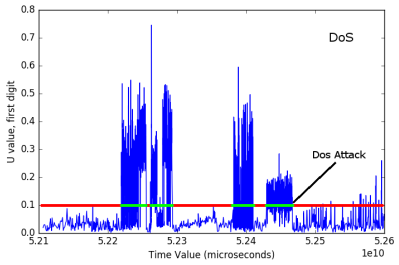


Figure 3: Benford's law analysis of packet inter-arrival times during four short DoS attacks

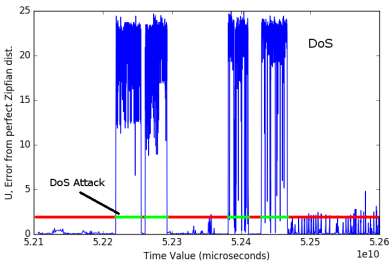


Figure 4: Zipf's law analysis of packet length during the scenario displayed in Figure 3

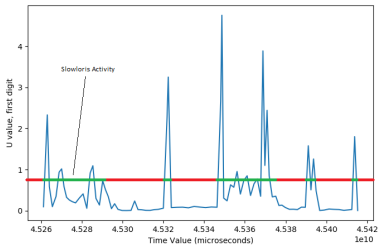


Figure 5: Zipf law analysis of packet length during a Slowloris attack

are power law probability distribution compliant, we will be discussing what is known to work and what is believed to work currently in research circles. We will discuss DoS detection and network anomaly detection methods we have looked at in our research, then we will quickly investigate the possibility of user activity profiling, and then finish up with our overarching detector system, the Probability Engine to Identify Malicious Activity or PEIMA, as an example of how to apply this to a network.

## 3.2 Denial of Service Detection

Power law probability distributions, especially Benford's law, are extremely effective at detecting flood based DoS attacks [10]. This is because flood DoS attacks, simply by virtue of being flooding attacks, break any conceivable naturalness of a network. Packet inter-arrival times conform to Benford's law on account of them being generated by that naturalness. Benford's law tends not to work if the numbers in a data set are not long enough, as Benford's law partially works on the interaction of orders of magnitude and natural proba-

bility distributions. Because flooding attacks are dependant on large quantities of data passing into a network to overwhelm the target, necessarily this must reduce the time between each packet. As a result the traffic cannot follow Benford's law. In order to reverse this, the traffic must be spread out again to follow a more natural rate. At worst, this is equivalent to heavy user usage, and is no longer an effective DoS. Intriguingly, even very heavy user usage follows power law probability distributions in general naturally. This means that power law probability distribution analysis is not only difficult to fool, it is capable of determining the difference between suddenly heavy user traffic and a DoS attack by virtue of the laws themselves.

Denial of service attacks appear as distinct deviations from conformity with Benford's law as can be seen in the multiple attacks in Figure 2. This appears with either Benford's law applied to inter-arrival time (Figure 3), or Zipf's law with whole packet length (Figure 4). Zipf's law and packet length tend to work due to flood based attacks having similar packet sizes from their attackers, especially in cases of amplification attacks, where the attack is often tuned to the most effectively amplifying response. This also allows us to determine something about the attack at the same time as detecting it. A weak or non-existent anomaly in length when analysed by Zipf's law may indicate an attacker aggressively randomising packet sizes, or many attackers with unique packet sizes, while a strong anomaly may tell us that an attacker is using the same packet for the attack.

This ability to learn something about attacks at the same time they are detected can theoretically be extended to other metrics. An anomaly in TCP flag types (SYN, ACK, etc) when analysed would also indicate a flood, and would be significantly stronger in the case of a single flag based flooding attack (SYN flooding, for example). A lack of such anomaly whilst still detecting anomalies in length or inter-arrival time could indicate that the attack was over UDP, and so on.

We can also detect attacks that are not strict flooding attacks. In Figure 5, we can see the results of power law analysis of packet length during a Slowloris attack. While it is not consistent, we can still see large and distinct anomalies from power law conformity, and therefore know an attack is underway. Consequentially, as this looks markedly different from traditional flooding based attacks, which are consistently large anomalies over time, one would be able to tell that an attack is a Slowloris attack simply by looking at it.

Power law probability distributions work generically across network traffic. This means that any sufficient subset of traffic can be analysed in the same way that traffic in general can be. The result of this is that if one splits up the network traffic based on source or destination, the apparent source and target of an attack can be determined due to the presence of anomalies matching those in the overall network traffic. This also means that we can detect both DoS and

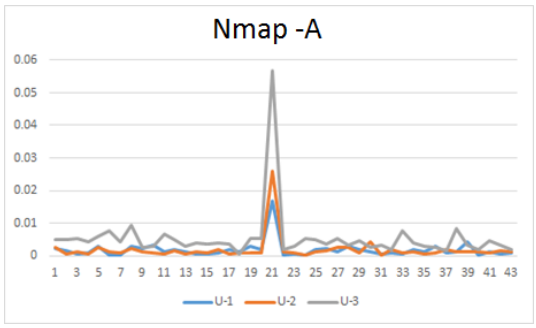


Figure 6: Three digit Benford's law analysis for an Nmap session using the -A flag

Distributed DoS (DDoS) attacks, as cumulatively, all traffic can be analysed in the same way. Even if an attacker performs a timing attacks such that each attacking client is not performing a DoS alone, we can analyse all of their traffic together and still detect an ongoing attack.

Essentially, with only extracting packet length and the time stamp from each packet header, one can detect DoS attacks and learn something about them. Collecting more information from the header (TCP flags, source and destination IP addresses) allows us to learn even more. Notably, this requires no learning, no deep packet inspection, is hard to fool, and even works while the contents of the packets are encrypted.

### 3.3 Network Anomaly Detection

Power law probability distribution based detection methods are not limited to DoS attacks, and have potential toward general intrusion detection [4, 5]. What we know so far is that attacks based on rapid, or similarly sized packets can be detected for similar reasons to DoS attacks. An example of this kind of attack is CVE 2013-2028, a brute force buffer overflow style attack which attacks the server very quickly with a rapid succession of packets. Another example is Nmap's -A profile, as can be seen in Figure 6. Anomalies generated this way are smaller and less easy to spot than DoS generated anomalies, however they are still present. While it is feasible that some attacks will have more recognisable anomalies than others, scripted attacks that happen over time will naturally distort a network in different ways to an attack that happens very quickly. It is likely that the best way to handle such classification is with some kind of signature based or learning agent. This agent would not have to learn the nature of a network though, that is already handled by the applicability of power law probability distributions. The agent would instead be designed to detect anomalies in a network and determine what kind of attack it may be. Such a system does not exist yet, so the effectiveness of such a system cannot be guaranteed. That

said, research into power law probability distribution based network anomaly detection outside our own has shown great promise, and as a completely different way of looking at network data presents further opportunities for detecting attacks that may have been missed by other solutions on a network.

## 3.4 User Profiling

Networks are not the only field that power law probability distributions can be applied. It has been shown that power laws are also effective at determining user identity, specifically through keystroke profiling [5]. Essentially, keystroke latency follows Zipf's law, allowing us to know at a glance whether a user is actually typing. This may not seem helpful at a first glance, however, attacks using malicious USB devices that present themselves as keyboards to hijack a computer would be immediately detectable via this method, as would any script or program that acts as keyboard input. What is far more interesting however is that individual typists differ in different ways from Zipf's law. One typist will have a different set of latencies to another, and will be different from Zipf's law in unique amounts to other users. This allows a profile of that user to be built, and for deviations from that user's profile to be detected. In this way, one could determine when a user's account was being used by an unauthorised user.

This concept of profile building can be extended. It was also shown in [5] that while Benford's law does not apply to keystroke latency, it can be used to determine a profile in the same way Zipf's law can. This is because while the latency is not necessarily a number generated in such a way that conforms to Benford's law, the use of said law allows one to detect when the system generating that number (the user) has changed. This can be theoretically extended out to any user affected process. One could feasibly profile user commands, user network usage, and other user computer usage to profile a user and determine when their activity has changed.

## 3.5 PEIMA system

PEIMA, the Probability Engine to Identify Malicious Activity, is a framework we have built to showcase what we would expect a power law probability distribution detection engine would look like. The engine consists of a number of detectors, each providing a stream of conformance datapoints to a decision engine, that then uses those datapoints to determine if there is an attack, and what measures to take to respond to it. For example, in a DoS scenario, the engine (as displayed in Figure 7) would detect a strong, continuous anomaly in both zipf analysis of length and benford analysis of inter-arrival time, and could send an alert to security staff, and maybe even attempt to alter the network gateway to drop packets coming from the apparent attacker. While this sounds a lot like a usual IDS/IPS style set up, the difference is that power law analysis is so light weight, this could very easily be entirely deployed on the gateway itself.

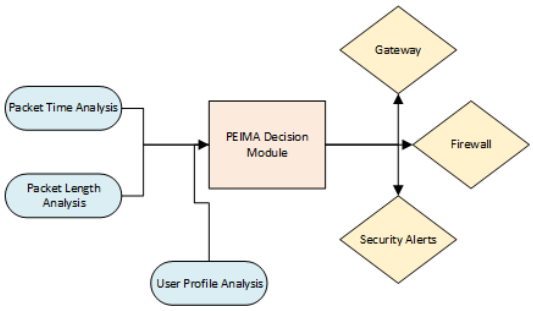


Figure 7: Example overview diagram of PEIMA system

Due to the inherent capability for Benford's law to be hardcoded into a detection mechanism, as well as the minimal amount of mathematical work to determine if an anomaly exists or not, and finally the fact that detection can be as simple as a numeric threshold, PEIMA can run on very low resources. That does not mean that PEIMA is limited by low resources, a far more complicated example could be a system integrated into an existing security setup.

Existing security appliances, for example network taps and log analysis software currently feeding a Security Information and Event Management (SIEM) style solution could also perform power law probability distribution analysis, or have such analysis performed on their exported data. PEIMA's decision mechanism could then be integrated into the existing alert generating software as another view on the data collected from around the network. This could allow for more accurate alert generation, as by virtue of power law probability distribution conformance being hard to fake, an alert for an attack with corresponding PEIMA alerts would be more likely a legitimate attack than one without. Also, suspected infiltration of a network coupled with deviations from PEIMA user profiles would be more severe than without.

PEIMA could be implemented as a host based IDS, or a network based IDS, or both. This flexibility, low overhead, and lack of training time for what is effectively an anomaly based IDS system is what can set the PEIMA based system apart from other anomaly IDS solutions.

# 4 Conclusions

Power law probability distribution based anomaly detection provides a new way of looking at information security metrics in general. In the realm of network security, it has been shown to be capable of providing a new, fast, and accurate way of detecting DoS attacks, as well as having possible extensions to general intrusion detection across a network. It is also possible to extend such anal-

ysis to user profiling and host based intrusion detection. In this white paper, we have discussed some of the required details in constructing such a system, as well as some of the inherent benefits of such a system. We have also discussed a framework, PEIMA, around which one can build a power law based anomaly detection system. Power law probability distribution analysis is new to information security, and the possible benefits and applications are not yet fully plumbed, so expect not only to see these starting to appear in solutions in the near future, but also to see more applications in the security space.

# References


[1] K. Angrishi, "Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets," *arXiv preprint arXiv:1702.03681*, 2017.

[2] L. Arshadi and A. H. Jahangir, "Benford's law behavior of internet traffic," *Journal of Network and Computer Applications*, vol. 40, pp. 194–205, 2014.

[3] C. Durtschi, W. Hillison, and C. Pacini, "The effective use of Benfords law to assist in detecting fraud in accounting data," *Journal of Forensic Accounting*, vol. 5, no. 1, pp. 17–34, 2004.

[4] F. Gottwalt, A. Waller, and W. Liu, "Natural laws as a baseline for network anomaly detection," in *Trustcom/BigDataSE/I SPA, 2016 IEEE*. IEEE, 2016, pp. 370–377.

[5] A. Iorliam, "Application of power laws to biometrics, forensics and network traffic analysis," Ph.D. dissertation, University of Surrey (United Kingdom), 2016.

[6] M. Lesperance, W. Reed, M. Stephens, C. Tsao, and B. Wilton, "Assessing Conformance with Benfords Law: Goodness-Of-Fit Tests and Simultaneous Confidence Intervals," *PloS one*, vol. 11, no. 3, 2016.

[7] S. Mohurle and M. Patil, "A brief study of wannacry threat: Ransomware attack 2017," *International Journal*, vol. 8, no. 5, 2017.

[8] I. Moreno-Sánchez, F. Font-Clos, and Á. Corral, "Large-scale analysis of Zipf's law in English texts," *PloS one*, vol. 11, no. 1, p. e0147073, 2016.

[9] L. Pietronero, E. Tosatti, V. Tosatti, and A. Vespignani, "Explaining the uneven distribution of numbers in nature: the laws of Benford and Zipf," *Physica A: Statistical Mechanics and its Applications*, vol. 293, no. 1, pp. 297–304, 2001.

[10] S. Prandl, M. Lazarescu, S. Soh, D.-S. Pham, and S. Kak, "An investigation of power law probability distributions for network anomaly detection," in *Proceedings of the 2nd International Workshop on Traffic Measurements for Cybersecurity*, 2017.

[11] B. Stanton, M. F. Theofanos, S. S. Prettyman, and S. Furman, "Security fatigue," *IT Professional*, vol. 18, no. 5, pp. 26–32, 2016.

[12] G. Weaver, A. Furr, R. Norton *et al.*, "Deception of phishing: Studying the techniques of social engineering by analyzing modern-day phishing attacks on universities." *Preprint*, 2016.