On the reliability of network measurement techniques used for malware traffic analysis

Joseph Gardiner and Shishir Nagaraja

University of Birmingham, B15 2TT, UK
{j.gardiner, s.nagaraja}@cs.bham.ac.uk

Abstract. Malware attacks are increasingly popular attack vectors in online crime. As trends and anecdotal evidence show, preventing these attacks, regardless of their opportunistic or targeted nature, has proven difficult: intrusions happen and devices get compromised, even at security-conscious organisations. As a consequence, an alternative line of work has focused on detecting and disrupting the individual steps that follow an initial compromise and that are essential for the successful progression of the attack. In particular, a number of approaches and techniques have been proposed to identify the Command & Control (C2) channel that a compromised system establishes to communicate with its controller. The success of C2 detection approaches depends on collecting relevant network traffic. As traffic volumes increase this is proving increasingly difficult. In this paper, we analyse current approaches of ISP-scale network measurement from the perspective of C2 detection. We discuss a number of weaknesses that affect current techniques and provide suggestions for their improvement.

1 Introduction

Malware detection and mitigation is a significant security challenge. In the last several years, the number of attacks, their sophistication, and potential impact have grown substantially. On one hand, opportunistic attacks have continued to flourish: these attacks are financially motivated, are responsible for the compromise of large numbers of machines, and result in the stealing of financial data, such as credit card numbers and online banking account credentials [9].

At the same time, targeted attacks have emerged as a new threat. These attacks target specific organisations or individuals with the intent of obtaining confidential data, such as contracts, business plans, and manufacturing designs [13].

Statistics and anecdotal evidence indicate that *preventing* attacks, either opportunistic or targeted, is difficult. For example, news reports have indicated that even security-conscious, well-funded organisations have fallen victims to attacks [12, 15, 16].

Considering the difficulties in effectively preventing attacks, defenders have looked at ways of *detecting* and *disrupting* the individual steps that follow an initial compromise and that are essential for the successful progression of an attack.

This is the so-called kill chain approach to defence [11]. In particular, considerable effort has been spent in identifying the establishment of Command & Control (C2) channels, i.e. the communication channel through which attackers control compromised devices and receive any data stolen from them.

Focusing on the C2 step has several advantages. It is a general, widely applicable measure, since both opportunistic and targeted attacks rely on the establishment of C2 channels. In addition, if defenders can detect the attack before sensitive data is ever ex-filtrated, the damages suffered by the attack's target are limited considerably. Even in the event of successful data theft, an understanding of the C2 structure could prove essential to determine what has been stolen and where it ended up. Furthermore, the analysis of the C2 channel may provide indications useful to attribute the attack to specific groups of people, which may facilitate legal actions against them.

However, C2 channel detection techniques make a critical assumption. They assume the existence of a measurement system that collects traffic containing C2 and non-C2 traffic. As traffic volumes increase, the measurement goal of storing all traffic becomes increasingly difficult. Core routers operate in the order of 100Gbps and are expected to increase to 1Tbps in a few years time. Enterprise routers are expected to scale up similarly, from 10Gbps to 100Gbps. At these throughput rates on a per-router basis, storing all traffic for a few days is a task that is practically impossible.

The scope of this paper: is to conduct the first security analysis of traffic measurement mechanisms, specifically those which C2 detection techniques depend upon. The research question we examine is: how hard is it for malware to evade current measurement mechanisms?

The main insight of our work is that all major sampling-based measurement mechanisms have security vulnerabilities. Thus we can expect that in the near future, malware designers will exploit these vulnerabilities to evade measurement and collection. Thus malware detection based on statistical pattern analysis of command and control traffic could be rendered useless.

2 The Command and Control Problem

Command and Control identifies the step of an attack where the compromised system makes contact back to the attackers to obtain additional attack instructions and to send them any relevant information that has been collected up to that point. It is one of the phases of malware intrusion. There are several others which we document briefly, as follows.

In the *reconnaissance phase* the attacker learns more about its target and identifies the weaknesses that will be exploited during the actual attack.

In the *initial compromise phase* the attacker attempts to compromise the network via various methods: spear phishing [20], social malware [14], or a "watering hole" attack – a opportunistic drive-by-download attack [18], in which victims are attracted, by different means, to a malicious web page. If successful,

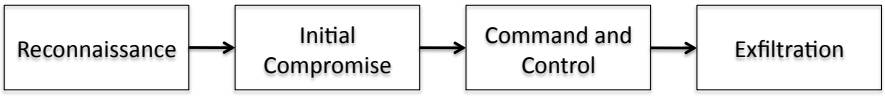


Fig. 1: The attack life cycle

the exploit downloads malware on the victim's machine, which as a consequence, becomes fully under the control of the attacker [17].

In the *Command & Control phase*, the adversaries leverage the compromise of a system. More precisely, compromised systems are forced to establish a communication channel back to the adversary through which they can be directly controlled. The C2 channel enables an attacker to establish a "hands-on-keyboard" presence on the infected system (via so-called remote access tools), to install additional specialised malware modules, and to perform additional malicious actions (e.g. spread to other machines or start a denial of service attack).

In the *exfiltration phase*, the attackers extract, collect, and encrypt information stolen from the victim's environment. The information is then sent to the attackers, commonly through the same C2 channel that was established earlier.

In the following sections, we analyse existing measurement techniques that can be applied to detection and disruption of targeted attacks, with a view to understanding unsolved challenges and open problems.

3 New attacks on Network Monitoring

Recording complete traces will prove increasingly difficult. Enterprise networks carrying a few tens of terabytes a day could result in hundreds of gigabytes of traces. It's currently possible to store traffic header traces for a few days at a time. However, the growth in network speeds will change this for the worse in the future. In the case of ISPs, the volume of traffic flow records is immense. A tier-1 ISP carries close to a hundred petabytes of user traffic per day [1], resulting in hundreds of terabytes of traffic header traces. Even with low storage and transmission costs, storing entire traffic traces beyond a short period of time is not feasible for ISP traffic while storing the entire traffic including packet data is outright impossible.

Currently, traffic monitoring is performed by routers, commonly using the Netflow [4] or sFlow feature. Alternatively, standalone measurement devices [6] observing traffic via network mirroring devices or splitters (optical or electrical) are more flexible than in-router methods. In both cases, traffic traces are exported to collectors which store the traces.

Network monitoring needs to guarantee that C2 traffic will be recorded. This can be hard to achieve as the malware can transform behaviour to evade measurement, as a consequence detection fails. Therefore apart from scalability, network

measurement systems must address evasion resilience for which network-wide control is necessary.

The goal of monitoring is to accurately estimate statistical quantities of relevance to the detection algorithms. We show that it is possible to compromise the reliability of monitoring techniques with fairness guarantees by targeting their *estimation accuracy*. The relevant metric for analysing measurement evasion, is the upper bound on the variance of estimation accuracy. We will consider two sampling methods (**uniform sampling** and **weighted sampling**) and two notions of fairness (**Max-min fairness** and **Proportional fairness**).

3.1 Attacks on uniform sampling

Uniform sampling involves sampling each event with equal probability. A monitoring system based on this approach is Sample and Hold [8]. Uniform sampling is also used in Netflow and sFlow measurement methods which are in current deployment in most routers.

Feature distribution attack (passive): C2 traffic can easily evade uniform sampling by modifying distributions of relevant traffic features. Uniform sampling is particularly ill-suited for estimating power-law distributions. A power-law distribution is of the type $f(x)\ x^{-\gamma}$. Thus sampling with uniform probability p will mostly obtain samples representing the majority while C2 traffic escapes in the tail-end.

Spurious flow attack (active): As a variant of this attack, an active attack can be carried out by inducing a few large legitimate flows which increases the probability that a majority of the recorded packets belong to the induced flows. Note this does not require a DoS attack against any router.

To estimate the value of a traffic feature x, applying elementary sampling theory [10], an unbiased estimate for feature x is $o_x n/k$, where n/k is the uniform sampling rate and o_x is the number of observations of x within the sample set. The accuracy of the estimate is given by its variance
$o_x n/k(n/k-1)$. Increasing n by a factor of n' (active attack) decreases accuracy by a factor of n'^2. Thus most of the packet sampling budget is spent on large flows, allowing low-volume C2 flows to go systematically undetected.

3.2 Weighted sampling

Weighted sampling [7] addresses the underlying bias of uniform sampling to accurately record traffic features used by detection algorithms such as byte count of a traffic flow. It does so by preferentially sampling from relevant traffic subpopulations. For instance, preferential selection of long-lived flows enables accurate byte count. Without fairness guarantees, weighted sampling is also damaged by the same attacks as uniform sampling, although to a lesser extent.

Given a set of n flow records with byte count $\{x_1, x_2, \ldots, x_n\}$, sampled independently, the goal is find the best sampling function $p = \{p_1, p_2, \ldots p_n\}$.

Applying Horvitz-Thompson [10], an unbiased estimation of each x_i is given by (1).

$$x'_i = \begin{cases} \frac{x_i}{p_i}, & \text{if at least one sample of } x_i \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

This formula is used by all sampling methods and is a root cause of monitoring problems, as we shall see.

The variance in estimation accuracy is given by: $Var(x'_i) = x_i^2 \left(\frac{1}{p_i} - 1\right)$. Finally, sampling theory introduces a cost function. By optimising different cost functions, we can generate different sampling methods.

Proportional fairness: counters the weaknesses of uniform sampling that allows flooding attacks, by sampling packets with an Inclusion Probability Proportional to Size (IPPS). Thus $p_i = c_i / \sum_i c_i$. The resulting inclusion probability (into the sample set) is $\pi_i = 1-(1-p_i)^k$. Proportional fairness allocates sampling budget in proportion to the data rates of different traffic sub-populations.

Max-min fairness: A sampling budget allocation is fair if there is no way to increase the budget of any traffic sub-population without decreasing the allocation another sub-population. Max-min fairness is trivial to evade once the allocation is known, by applying the intelligence variance attacks. Uniform allocation is equivalent to uniform sampling and its attacks.

Minimal cost sampling method [7]: Minimises cost function $\sum_i (x_i^2/p_i + z^2 p_i)$ subject to $p_i > 0$, where z is the sampling threshold. Small flows $x_i < z$ are IPPS. Large flows $x_i \geq z$ are included with probability 1 (up to maximum sampling budget).

Proportional sampling method: Combines proportional fairness maximising a cost function of the sum of the logarithms of the allocated sampling budgets.

Other methods: There are three other methods which improve upon the ones above. VarOpt sampling [5] uses both IPPS and max-min notions of fairness. It has a fixed sampling budget (selects exactly k items on average $\sum_i p_i = k$). It minimises the cost function minimising variance across traffic sub-populations.

Attacking proportional fairness: Having discussed weighted sampling, we now propose an generic attack on all sampling techniques based on proportional sampling: proportional fairness, IPPS, and VarOpt sampling. Given the observed samples S, the information theoretic uncertainty (entropy) of the distribution over flows is the sum of two parts: $H = -\sum_{i \in S} p_i \log p_i - \sum_{i \notin S} p_i \log p_i$. One part is the contribution of observed samples and the other is the contribution of unobserved packets. Thus an error due to the **contribution of unobserved sub-populations** adds up to $-\sum_{i \notin S} p_i \log p_i$.

This could be far from negligible and its size depends on the p_i for $k \notin S$. This error arises due to weight-based inclusion where rare packet constructions have extremely low sampling probabilities. To exploit this vulnerability, we can apply the signature evasion attacks where C2 traffic continually changes form in a random manner, thus entering the noise floor as far as sampling is concerned, contributing to a negative entropy balance.

3.3 Network-wide orchestration

Secure measurement techniques require more than just data collection. Since sampling budgets are fixed, attackers can exploit this by flooding the sampling buffers with random data. For instance, gradually increasing the bitrate of random transmission in advance of actual C2 transmission impacts both machine learning and measurement: a high-entropy traffic feature would program ensemble based approaches such as RandomForests to look elsewhere as well as overwhelming local sampling resources.

To address these requirements, we need network-wide orchestration of measurement resources combined with flexible sampling budgets at each router. Network-wide coordination between routers can leverage unused sampling budgets at one router to cover the overflowing sampling budgets at an upstream router. Similarly, in response to localised flooding, the router might wish to initiate a change in routing topology to improve measurement. There have been a few attempts at designing such systems but these are very early days. Optimal Network-wide Sampling [3] tries to maximise the probability that every flow is sampled at least at one of the routers; cSamp [19] coordinates sampling over multiple locations to avoid duplicate measurement.

The paradigm of software-defined networks (SDN) is ideally suited for implementing such systems. SDN de-constructs a hardware router into a software controller based on a general purpose computer and specialised packet-forwarding hardware. This separation allows rapid response to dynamic changes in network state, such as efficiently dealing with localised flooding attacks intended to overwhelm the measurement system. As an early example, OpenSketch [21] proposes an extensive framework for scalable and adaptive measurement based on the Software-Defined Network paradigm. OpenSketch seeks to achieve a optimal coordinated measurement in response to network events. It uses fast hardware primitives to drive coordinated measurement of statistically significant traffic across wide-scale networks, but does not support the monitoring of low-volume traffic sub-populations such as C2 channel traffic.

4 Evaluation

To evaluate the different sampling methods, we apply each method to the CAIDA UCSD Internet Traces 2012 dataset [2]. This dataset contains anonymised internet traces captured passively on high-speed internet backbone links. We then sample the data according to flow size, use the output of each sampling method to estimate the ground truth, and measure the amount of lost information (in terms of estimation error). A positive error means an overestimation from sampling, and a negative error shows underestimation. Negative error can be seen as worse, as it indicates that significant information has been lost. The ground truth of flow sizes in the dataset can be found in Figure 2. The majority of flow sizes are in the range of 50 to 100, with a maximum of 298500 (resulting in a mean of 441). We evaluate four sampling strategies on the dataset: uniform, proportional fairness and inverse proportional (computed as $1 - p$, where p is

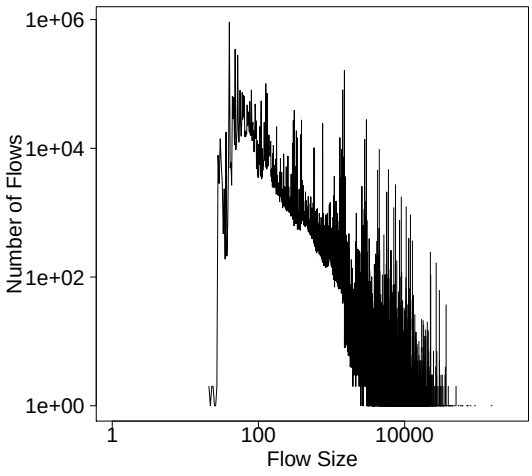


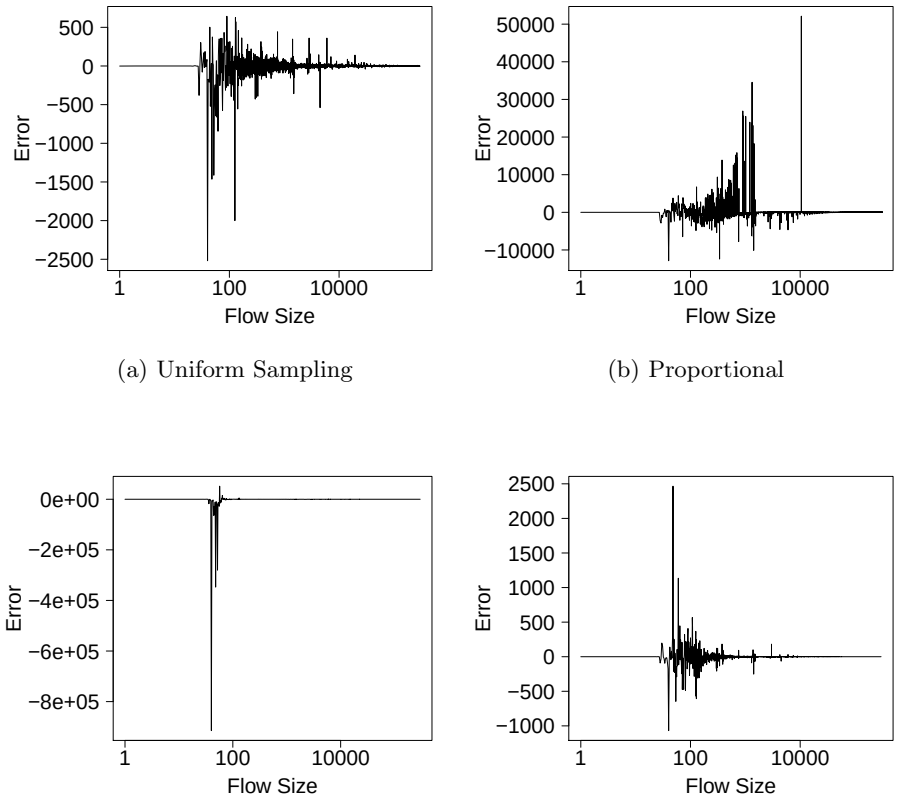
Fig. 2: Distribution of flow sizes in dataset

the probability density function used be proportional fairness). The fourth sampling strategy is one of our own design called threshold inverse sampling, which we describe below, and is used to test if sampling can be improved by using a combination of the previous strategies.

Threshold Inverse Sampling We propose a simple sampling strategy that could be used to evaluate the possibility of improving the accuracy of the sampled output. For this, we suggest threshold inverse sampling. In this approach, traffic is sampled using the inverse proportional fairness method, i.e. the sampling probability is inversely proportional to the event probability. This ensures that rare events are almost always captured. To avoid the possibility of an adversary "hiding" within frequent events, where the probability of being sampled is low, we also apply uniform sampling. By doing this, the probabilities for common flow sizes are raised to a threshold, meaning that all traffic has a minimum probability of being sampled. This level can be set appropriately. In part this will define the overall amount of data to be sampled as this will represent the most common events which will make up the largest proportion of sampled data. While this is not proposed as a full solution, we use it to measure if improvements can be made by combining sampling approaches into a single approach. As with the previous approaches, this is a static sampling strategy.

4.1 Results

Figure 3 shows the amount of error observed by the estimation of the ground truth from the various sampled outputs. The error is computed as the estimation



(a) Uniform Sampling

(b) Proportional

(c) Inverse Proportional

(d) Threshold Inverse

Fig. 3: Error of estimation from sampling methods

minus the ground truth, which results in the estimation error. Figure 3(a) shows that uniform sampling results in error across all flow sizes. While the error is relatively minor for frequent sizes, for rarer sizes (small and large flow sizes) the error is significant. As the error values are largely negative, this indicates significant information is being lost. This is down to rare flow sizes being sampled with the same probability. In this case the probability is set to 0.2, so those flow sizes that feature less than 5 times are unlikely to be captured.

Proportional fairness, as shown in Figure 3(b), achieves a slightly higher error rate for common flow sizes, but performs comparatively worst for rare sizes. Again, this is down to rare events being assigned a much lower probability than frequent events, meaning they are very unlikely to be captured. Common

events are over-sampled resulting in positive error amounts, while rare events are under sampled, or in most cases not at all, resulting in negative error.

Inverse proportional sampling solves this problem by assigning the highest probabilities to rare events, meaning that they are almost always captured. The main downside is then the reverse or proportional sampling – frequent events are assigned a low probability and are therefore not sampled as accurately, as illustrated in Figure 3(c). In this case, the frequent flow sizes result in an under-estimation error of up to 80000 for the common sizes, indicating severe under sampling of flows. There is also error on the middle-ground flow sizes of up to a few hundred under estimation.

Finally, the threshold inverse method (figure 3(d)) that we have put forward features the best accuracy across the entire distribution of flow sizes. In this experiment, the threshold was set to 0.2, applied to the inverse proportional density function. The method samples common events with a high accuracy (less than 2500 errors), and accurately samples the rare events (events that occur a limited number of times feature, on average, an error of <0.0001). The error is relatively evenly distributed between positive and negative values. The error for common events is on the same scale as uniform sampling, which is to be expected.

4.2 Discussion

Our experiments demonstrate that the major sampling methods (Uniform, Proportional, and Inverse Proportional) exhibit vulnerabilities in the form of sampling bias. These biases enable adversaries to shape traffic behaviour to evade sampling. Even the threshold approach (proposed by this paper) that attempts to prevent this by ensuring all behaviour has a minimum likelihood of being sampled still loses information. Flows of common sizes only have a one in five chance of being sampled. Both uniform and proportional sampling poorly represent less common behaviour. In any sampling strategy that is used for detection these rarer events are useful so should always be sampled. The common behaviour cannot be ignored as it also provides valuable information and could contain malicious behaviour. All of these static approaches have to trade off collection on one type of traffic over another in order to maintain the effect of sampling, so each provides means for an adversary to hide.

The threshold sampling method will work well in cases where common events are large in number, thus need to be sampled to limit the amount of data, whilst maintaining accuracy, in particular for rare events. This is down to the fact that rare events will almost always be sampled, while the common events can be sampled to the desired rate by setting the threshold to an appropriate value. The distribution of events, such as flow sizes, should have tall, narrow peaks (power law and log normal distributions can provide this). Where it will not work so well, however is if the distribution features wide peaks with gradual slopes (such as a normal distribution).

A common limitation that all of these strategies, including threshold inverse, fall victim to is that they are all static in nature. They all assume constant

behaviour and do not take into account that an adversary can change their behaviour. They are also static in terms of the probabilities assigned for sampling, no matter how much traffic there is to be measured.

It is clear from this that dynamic strategies need to be developed in order to maximise the effectiveness of sampling given the current state of the network. A simple solution could be implemented by changing sampling strategies regularly in an unpredictable manner, with regards to the adversary. So, for one time period use proportional sampling, then for the next use inverse proportional. This would make it difficult for the adversary to shape traffic in order to evade sampling.

A point to consider is that the amount of traffic flowing through a network will not be static throughout the day. For example, a corporate network will have far less traffic outside of business hours than during. It does not make sense to keep the same sampling strategy in place during this time, when a greater percentage of the traffic can be collected and processed. So a sampling strategy should be in part influenced by traffic characteristics (such as cumulative traffic) other than that which is the key measure for sampling (such as the flow size), and be adjusted regularly.

With increases in computational power it may also become possible to apply in-line analysis to exclude certain flows from being sampled. Allowing resources to be used on more interesting flows. Of course this would have to be an extremely lightweight system that could, for example, recognise known legitimate web requests (for example, those to `bbc.co.uk` which are highly unlikely to be malicious). This solution will need to be carefully designed however; to avoid opportunities for an attacker to hide themselves. Rather than simply exclude flows from being sampled, flows to known safe locations (such as `bbc.co.uk`) could be sampled at a lower rate than to those domains that are untrusted or provide greater opportunity for abuse (such as `twitter.com`).

In larger networks that feature many different entry points to the network, as well as various levels of sub-nets within, sampling can be carried out at multiple points over the same traffic, using the same sampling strategy. This will provide more opportunities to capture malicious traffic while not increasing the strain on resources on any one point.

5 Conclusion

Behavioural analysis techniques depend on traffic collection mechanisms to detect malice or anomalies in network traffic. However to deal with high traffic volumes whilst ensuring low processing latency, network operators rely on measurement techniques that record a subset of the traffic, as opposed to recording full traffic traces. In this work, we have analysed the resilience of current network measurement techniques against intelligent adversaries that shape network traffic with the intent of evading collection.

One insight of our work is that current network measurement techniques are easy to evade. They exhibit biases that are readily exploitable. The second

insight is that it is possible to do better; we have proposed a new measurement technique that has better evasion resistance properties. However, it is far from perfection and further investigation into evasion resistant sampling techniques is necessary.

Both academia and industry have been fighting malware C2 communication channels for close to a decade now. The focus of most of the current work is in the direction of detection mechanisms. However little attention has been paid to measurement techniques – the assumption that all traffic can be recorded is increasingly under stress as traffic volumes increase. In this context, reliable measurement techniques are an important requirement. If C2 traffic cannot be recorded, then detection algorithms cannot work, no matter how good they are.

From time to time, experts have proclaimed that the problem has been solved, only to find their confidence has been misplaced due to subsequent attacks. In this paper we argue that a focus on global-scale measurement architectures for C2 traffic is missing and needs attention. Current ISP-scale measurement techniques do not offer the properties necessary for C2 detection, thus creating a gap where surveillance apparatus can work without encumbrances.

Thus an important challenge is the scalable collection of traffic traces. With increasing traffic rates it will soon become hard to store all traffic even in enterprise networks thus forcing defenders to rely on estimation via sampling techniques. This is already required at ISPs and datacentre networks.

In the light of these challenges, the problem of characterising C2 traffic behaviour from sampled traffic requires a shift of perspective. Researchers need to take a step back to focus on the big picture. First, the challenges of building secure measurement techniques has not received the necessary attention in the security community — the 'needle-in-the-haystack' problem is challenging and some approaches have been outlined from sampling theory but these do not work in an adversarial setting. Apart from sampling techniques, the measurement architecture has to be open and extensible, allowing network wide coordination to focus measurement resources on attack traffic rather than trying to work out broad trends as it has historically done.

There is a pressing need for the research and development of better publicly available C2 defence techniques, especially built into routers, which are essential to routing information, and where data naturally aggregates. The need for open and flexible frameworks might benefit from Software-Defined Networking. These open-source platforms are of great value. SDN deconstructs current hardware routers into controllers (running on general purpose computers) and programmable hardware running on specialised hardware. This allows incorporating innovations in sampling and detection directly into the router. SDN massively slashes the costs of evaluating and deploying techniques. Expensive hardware routers costing hundreds of thousands of dollars, are replaced by general-purpose computers and a one-off investment in routing hardware. Thus enabling rapid deployment of new techniques that keep up with attacker advances.

The increasingly targeted nature of todays attacks are indicative combined with high levels of attacker motivation presents a challenging problem. Judging

from innovations in targeted malware, we see the need to develop traffic measurement mechanisms which can accurately instrument traffic characteristics of malware with high-stealth properties. Since C2 is a critical part of malware design, we expect malware capabilities to shape and morph traffic in orer to achieve full measurement evasion. We hope that this paper will help with the development of novel measurement techniques which can keep up with malware agents that incorporate dynamic traffic morphing behaviour.

References

1. AT&T global networking facts. `http:www.corp.att.com/gov/about_ags/fact_sheet`.
2. The CAIDA UCSD Anonymized Internet Traces 2012. `http://www.caida.org/data/passive/passive_2012_dataset.xml` (accessed on 2013–03–20).
3. G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot, and P. Thiran. Reformulating the monitor placement problem: Optimal network-wide sampling. In *Proceedings of the 2006 ACM CoNEXT Conference*, CoNEXT '06, pages 5:1–5:12, New York, NY, USA, 2006. ACM.
4. Cisco Systems Inc. Cisco IOS Netflow. `http://www.cisco.com/web/go/netflow`.
5. E. Cohen, N. G. Duffield, H. Kaplan, C. Lund, and M. Thorup. Stream sampling for variance-optimal estimation of subset sums. In C. Mathieu, editor, *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 1255–1264. SIAM, 2009.
6. C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk. Gigascope: a stream database for network applications. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 647–651, New York, NY, USA, 2003. ACM.
7. N. Duffield, C. Lund, and M. Thorup. Learn more, sample less: control of volume and variance in network measurement. *IEEE Transactions on Information Theory*, 51(5):1756–1775, May 2005.
8. C. Estan and G. Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Trans. Comput. Syst.*, 21(3):270–313, Aug. 2003.
9. J. Franklin, V. Paxson, A. Perrig, and S. Savage. An inquiry into the nature and causes of the wealth of internet miscreants. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 375–388, New York, NY, USA, 2007. ACM.
10. D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):pp. 663–685, 1952.
11. E. M. Hutchins, M. J. Clopperty, and R. M. Amin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Technical report, Lockheed Martin Corporation, 2010. `http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf`.
12. B. Krebs. Security Firm Bit9 Hacked, Used to Spread Malware. *Krebs on Security*, February 13 2013. `http://krebsonsecurity.com/2013/02/security-firm-bit9-hacked-used-to-spread-malware/`.

13. Mandiant. APT1: Exposing One of Chinas Cyber Espionage Units. Technical report, 2013. http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf.
14. S. Nagaraja and R. Anderson. The snooping dragon: social-malware surveillance of the tibetan movement. Technical Report UCAM-CL-TR-746, University of Cambridge, March 2009.
15. E. Nakashima. Confidential report lists U.S. weapons system designs compromised by Chinese cyberspies. *The Washington Post*, May 27 2013. http://articles.washingtonpost.com/2013-05-27/world/39554997_1_u-s-missile-defenses-weapons-combat-aircraft.
16. N. Perlroth. Hackers in China Attacked The Times for Last 4 Months. *The New York Times*, January 30 2013. http://www.nytimes.com/2013/01/31/technology/chinese-hackers-infiltrate-new-york-times-computers.html.
17. M. Polychronakis, P. Mavrommatis, and N. Provos. Ghost turns zombie: Exploring the life cycle of web-based malware. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, pages 11:1–11:8, Berkeley, CA, USA, 2008. USENIX Association.
18. N. Provos, M. A. Rajab, and P. Mavrommatis. Cybercrime 2.0: When the cloud turns dark. *Commun. ACM*, 52(4):42–47, Apr. 2009.
19. V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen. Csamp: a system for network-wide flow monitoring. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 233–246, Berkeley, CA, USA, 2008. USENIX Association.
20. TrendLabs APT Research Team. Spear-Phishing Email: Most Favored APT Attack Bait. Technical report, Trend Micro Incorporated, 2012. http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-spear-phishing-email-most-favored-apt-attack-bait.pdf.
21. M. Yu, L. Jose, and R. Miao. Software defined traffic measurement with opensketch. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, NSDI'13, pages 29–42, Berkeley, CA, USA, 2013. USENIX Association.