A Risk Based Approach To Limit The Effects of Covert Channels for Internet Sensor Data Aggregators For Sensor Privacy

Camilo H. Viecco
Department of Computer Science
Indiana University
cviecco <at> indiana.edu

L. Jean Camp
School of Informatics
Indiana University
ljcamp <at> indiana.edu

ABSTRACT

Effective defense against Internet threats requires data on global real time network status. Internet sensor networks provide such real time network data. However, an organization that participates in a sensor network risks providing a covert channel to attackers if that organization's sensor can be identified. While there is benefit for every party when any individual participates in such sensor deployments, there are perverse incentives against individual participation. As a result, Internet sensor networks currently provide limited data. Ensuring anonymity of individual sensors can decrease the risk of participating in a sensor network without limiting data provision.

Two contributions are made in this paper. The first is an anonymity mechanism to defeat injection attacks. This defense mechanism is based on economics rather than classic cryptographic protocols. The second builds on the foundations created by the first. It is the a proposal for randomized sampling of correlated sensory inputs to asymmetrically increase the cost of sensor identification for attackers without significantly reducing the quality of the published data.

General Terms

Algorithms,, Measurement, Design, Economics.

Keywords

Keywords are your own designated keywords.

1.INTRODUCTION

The problem of sensor anonymity is derived from a need to share data. Our solution is constructed upon a foundation of network protocol analysis, information theory, and economics, rather than cryptographic assurances of anonymity. We begin by describing Internet sensor networks, then provide a brief overview of previous work on anonymity-enhancing network protocols. We also define the limitations of previous approaches and illustrate the advantages of the proposed approach.

Conference'04, Month 1–2, 2004, City, State, Country.

After this high level introduction, we focus on probe attacks for various classes of sensor networks. This includes a high level description of how attackers use probe networks to obtain covert channels.

The third major section details our proposed approach. We conclude that in the daily operation of sensor networks, economic incentives, and information theoretic defenses that increase the cost to attackers can create an effective defense.

1.1 Internet Sensor Networks

Attacks can be roughly categorized into two groups according to their targeting strategy: directed attacks and undirected attacks. Directed attacks or targets of choice occur when attackers purposely mount an attack on a previously identified and selected organization. Undirected attacks or targets of opportunity occur when attackers are searching for some class of resource in order to exploit it. In undirected attacks, the location of such resource is of minor importance. Organizations usually have very different approaches to defending against these two types of attacks; thus, being able to distinguish them is extremely useful.

Differentiating between directed and undirected attacks requires information about the global state of the Internet as close as possible to real time. Data on network status enables administrators to classify threats as directed or undirected, and thus choose an appropriate defense. In addition, by indicating the breadth of an attack, the victim can identify possible allies and collaborative sources of information. Global data enables administrators to better respond to abnormal behavior in their own systems. However, as each network administrator can only know the status of the network under his/her control, data sharing is required to produce a global view. Cooke et al.[29] show evidence that distributed data sharing is inadequate as different address blocks observe different traffic patterns. Thus, even a large aperture sensor is inadequate for knowledge of the network state if it is located in a continuous address block. Widespread sensor placement is required to have a representative sample of the global Internet. It is not a surprise that multiple data aggregation services have emerged to provide such global view.

Aggregation services collect, transform and publish some summary of the information locally gathered by the sensors. Sensors are individual sources of local network status such as honeypots or IDS. Examples of such services/systems include the Internet Storm Center (ISC or Dshield)[16], the Worminator[15], Neti@home[12], myNetwatchman[10], CAIDA[7], the University of Michigan Internet Motion Sensor[17], and the US Department of Homeland Security PREDICT system[24]. All these aggregation services work in a similar manner: data from sensors are collected, filtered and published at a predefined rate[1] The rates vary between services from one hour to twenty-four hours. The scope of publication also varies, with Dshield publishing the least detailed data to the public at large and the UM Internet Motion Sensor publishing detailed data only to its members. These observations indicate the understanding of the existence of a trade-off between the value and the risk of data availability. There is a concern that more public and detailed data may be more useful to attackers than to defenders. Effective anonymization of data sources can mitigate this trade-off between empowering defenders and enabling attackers.

The relationships between sensors, aggregation services and users of the service are summarized in Figure 1.

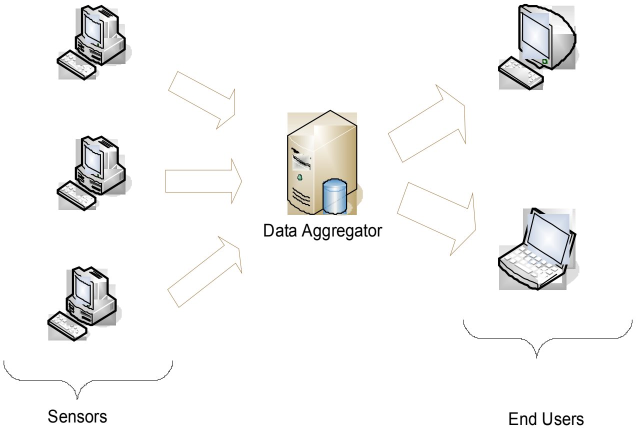

Figure 1. Data Flow For A Data Aggregation Service.

1.2 Previous Work

Maintaining source anonymity of widely published data has been a problem of interest in politics for several centuries[2]. The problem of measuring the efficacy of anonymization methods has two recent theoretical and practical contributions for measuring the efficacy of anonimization are important to this work. The first comes from Latanaya Sweeney [13,14], who not only reintroduced and analyzed the problem of cross-data identification, but also provided a solution for static data sets called k-anonymity. The second contribution comes from Serjantov and Danezis [22] who redefined the concept of 'anonymity set' in a more precise and information theoric manner. Serjantov and Danezis illustrated that several methods presumed to yield a high anonymity set provided much less anonymity than previously thought. While their work is based on mix networks their ideas can be expanded to other anonymity producing methods.

In the network security arena, the first efforts at providing methods for anonymity came from Flegel et. al[4,6]. Their efforts were directed at removing power from system administrators through anonymization of system logs. Minshall[9]; Fan et.al[5]; and Pand and Paxon[11] provided partial solutions to the problem of anonymization of IP addresses on captured packet traces. Slagel et. al [25] focused on the problem of netflow anonymization. Lakshmanan et. al[8] propsoed a generic transformation widely applicable to communication headers. Lincoln et al. [23] proposed a structure to enable sharing searching of IDS alerts in order to detect correlations. Unfortunately, with the exception of the packet traces anonymization methods and the works of Lakshmanan et al.[8] and Lincoln[23], the efficacy of the proposed solutions or methodologies have not been tested against data linking. In the case of packet traces, the possibility of cross data linking is made explicit but never analyzed.

Bethencourt et al.[2], was the first researcher to illustrate the problems of cross data linking in Internet sensors. The set of proposed solutions does not include measurements, nor does it provide theoretical bounds on the effectiveness of their solutions. This paper complements their work by providing a theoretical framework in which to address the problem of probe attacks as wel as giving potential solutions to a system with the parameters as Dshield.

Clayton et al.[28] makes a good introduction on the fallacies of some data anonymization systems. In particular, they conclude that: "... no operation concerning a pseudonym should have an observable side effect that could leak the identity of the user... ". Internet sensor networks, the domain of interest, are designed to show side effects. Yet the identity of internet sensors (the IP address), should remain hidden.

Another area of interest is the privacy preserving data mining. In particular, the work of Agrawal et al.[1], and Brickell and Shmantikov[26]. Their research is targeted on effectively anonymizing the sensors from data miners by using cryptographic or data perturbation techniques. We will explain what differentiates our work from previous work in section 1.4. We will provide details of the problem space, including the attacks models and trust assumptions, in section 1.3.

1.3. Defining the Problem

Our model assumes that the adversary has very little control over the network infrastructure, but does have complete control on many end points. We assume that the aggregation service is trusted by all the sensors, in that the aggregation service will not reveal the identity of the sensors. We assume that there is some mechanism to ensure that the communication channel between the sensors and the aggregation service is protected against traffic analysis. We assume that the aggregation service can uniquely identify any sensor with whom it has previously interacted. We assume that full aggregated data are available to the attacker (he/she belongs to the data sharing consortia), and that an attacker has control over some, but not a significant part of the sensors. We further assume that the sending of probes has a very small yet

[1] Actually, the DHS's PREDICT system would work on base of NDA agreements. It is still unclear of the need for a trust chain for researchers will be a limiter in the use of the data.

[2] Examples of anonymously published political documents include the Federalist Papers, and the translations of 'The Rights of Man and the Citizen', which were not welcomed by colonial powers at the end of the 18th century.

non-zero cost to the attacker. The problem that we are trying to solve is: Is there a way to make the probe sensor identification of a large portion of the sensors economically unfeasible? Can we provide a high lower bound on this cost? Further, Can we measure how much our data output changes when different mitigation mechanisms are applied? This last question will only be analyzed for a Dshield like system.

The assumption that an attacker is able to compromise multiple end points but not as likely to compromise infrastructure nodes is simply the recognition of botnets [21]. In our trust model, we trust the aggregation service, but do not trust the other entities that are also receiving the data from the sensors. This is consistent not only with botnets, but also with a grayhat adversary or adversaries that are competitors in other arenas.

One of the interesting elements of this problem is that attackers use the infrastructure, i.e. the reports of the sensor networks, to attack the infrastructure, the location and accuracy of the sensor network. This particular study focuses on adversaries that cannot control or observe how the information passes through the network, rather focuses on adversaries that take advantage of the implicit feedback loop generated by the process of publishing the data.

The key differentiators of the sensor network anonymization probse are: (i) the data are not static, data is periodically added to the output (ii) the data provided by the aggregator are available to the attacker, and (iii) the defender cannot distinguish 'a priori' probe data from bad injected data.

1.4 Comparison with Previous Approaches

With all the assumptions detailed above, it is reasonable to believe that this problem can be solved by applying previously published anonymization techniques. In this section we explain why some general techniques fail to address our problem.

Data filtering may seem like an obvious approach. The problem with data filtering is that abnormal network status data injected by attackers cannot be distinguished from abnormal network data due to non-probing attackers.

Mix networks or onion routing cannot be used as a defense mechanism against probe attacks (data injection) as these are designed to address a different problem. Mix networks and onion routing provide unlikable communication channels across untrustworthy communication intermediate peers that are trying to determine who is communicating with whom. In our solution and model, this part of the problem is assumed to be solved potentially by some implementation of these mechanisms such as Tor[27]. Further, our problem statement differs from anonymous communication problems in that our adversary has very limited control of the infrastructure, yet still controls many end points.

Sweeney's [13,14] emphasizes the use of k-anonymity only for static data sets. The process of re-identification of datasets is usually done with the use of external data utilized for cross data linking. For data that increases over time where the attacker has some control, another method can be used: the use of probe response attacks. The possibility of such attacks in the Internet has been known in the literature[11] but it was not until the work of Bethencourt et. al. [2] that an algorithm and simulations were published. Bethencourt et al. demonstrated the problem by showing how simulations allow easy discovery of sensors of the ISC[16]. In this paper, we generalize the costs for such identification procedures for any aggregation service in addition to providing guidance to mitigation mechanisms. The procedure we introduce increases the cost for the attacker while minimizing the distortion of the data released by the aggregation service.

The proposal of Agrawal et al. [1] consists of adding a random variable to the sensor data to effectively perturb the data output. If the random variable has a very large variance, this method requires a large number of inputs to effectively approach the original distribution. If the variance is small, the attacker need only to generate data outside the variance to create a reliably detectable signal.

The work of Lincoln[23] includes several techniques for anonymization and related defense mechanisms. The method they propose against probe response attacks is the use of randomized delay alert correlation, with the time stamp field scrubbed. This method cannot be reasonably used for our purposes, as data sharing for operational use requires near real time latency. Further, strategic (long term) use requires timestamps with at least a one day resolution.

The work of Brickell and Shmantikov[26] uses cryptographic techniques to unlink data thus protecting individuals from releasing their identity to data miners. However this work does not take into account the possibility that the data being reported can be influenced by the party that is trying to identify the identity of the data sources.

The approaches suggested by Bethencourt et al.[2], in particular the sampling of data outputs, appears to be a good compromise. In particular, Bethencourt uses economic incentives to prevent 'marking' of packets. The problem with this approach is that sampling is done on a per sensor level, after data have been collected. This approach does not increase the signal to noise ratio for the attacker. This approach does not work if we assume attackers with access to large botnets, as the defense mechanism leaves the attack trivially parallelizable.

All of the previously suggested techniques address the problem after the data have been aggregated. In economic terms, these post-collection sampling mechanisms provide more advantage to the attackers than the defenders. Post collection data transformation are more expensive for the aggregator than injection for the attacker, thus creating a systematic asymmetry. The approach presented here advantages the defenders by utilizing the ability to apply sampling at different dimensions and in different levels at event recording time. Thus attackers must synchronize their injected signals in all the possible filtering dimensions. The result is an economic disadvantage for the attacker, as described in more detail in the following pages.

2.PROBE ATTACKS AND INTERNET SENSOR NETWORKS

Internet sensor networks are the data source for Internet status aggregation services. Aggregation serves two functions: It centralizes data publishing, and enables limited anonymization of the sensors. Sensor anonymization is a fundamental requirement for the contributors as well as the quality of the aggregate data. An attacker who can identify the sensors will be able to: (i) hide attacks (hide or slow worm spread), (ii) hide a directed attack to an organization, or (iii) completely distort the quality of the exported data, thus making the data sharing effort useless.

Anonymization is so important, that despite economic benefits to data sharing[18], sharing detailed information security data is usually highly limited. Organizations that share internal data include the Honeynet Alliance [19] and the business sector-based ISAC structure in the US. But even within those groups, data are aggregated, filtered and thus transformed before release. (The Honeynet Alliance is a notable exception to this rule. Sensor anonymity is not an issue in the Honenet Alliance as the lifespan of honeypots is usually limited to a few intrusions). Many of the current data sources include sensors that are not easily relocated, such as Darknets. For others, the shared information is usually reduced to summaries of data for example as with the REN-ISAC[20].

2.1 Probe Internet Sensor Attacks

Probe sensor attacks use the feedback channel implicitly provided by the data compilation and aggregation. Since the publication phase cannot distinguish “good” users from “bad” users, a malicious user can send traffic into potential sensors to try to observe the abnormal signals in the output of the data aggregation. See Figure 2.

The most generalized statement of the problem from the attacker's perspective is: “Determine the parameters of an box with some controllable inputs and some observable outputs”. However the sensor identification problem differs from most system identification methods because our system has many inputs and outputs, and is generally non-linear. The attacker's objective is to estimate the sampling function used to collect the data from the Internet. The function’s secret parameters are the true location of the sensors, as the remaining parameters must be published in order to make sense of the published data.

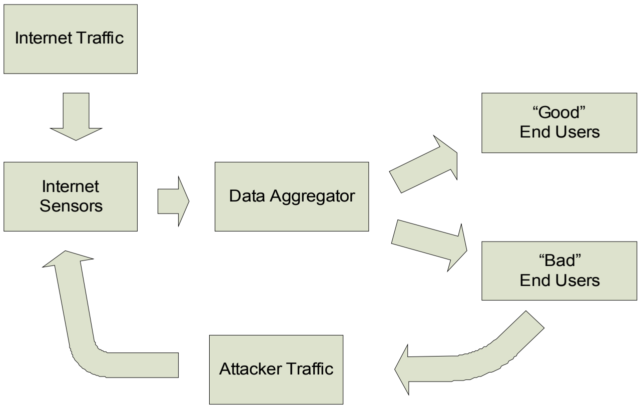

Figure 2. Data Flow For Probe Response Attacks.

The costs associated with running sensor identification attacks can be explained by running time and bandwidth costs. We will discuss two attack algorithms: a brute force approach and an N-ary recursive approach. These are analyzed in terms of their “running time”. This parameter is used to estimate the cost for an attacker. The running time of an algorithm describes a bound on the number of operations needed to complete the algorithm. The cost for each algorithm is expressed in terms of the needed bandwidth required for its operation.

2.1.1 Linear (Brute Force) Algorithm

This algorithm essentially iterates through each of the possible sensors to determine if it is a sensor or not. The algorithm is expressed below:

```
For each possible location
  Estimate the number of sensors in the current selected location
  if number of sensors is zero
      then discard location
  else location is a sensor
```

The running time of this algorithm is: U*K. Where U is the size of the search space and K is the number of iterations needed to determine whether a sensor has been located. In this case, a partition of size one. The bandwidth cost per iteration is P, where P is the number of packets required to generate a readable signal in the aggregate data. The total cost for such algorithm is calculated by multiplying the running time by the per iteration cost:

Total cost= running time * iteration cost= U*K*P.

This algorithm has a minimal cost, but also has a linear running time. A linear running time is unfeasible for large sensor spaces, such as the Internet.

2.1.2 N-ary Search Algorithm

Another way to approach sensor identification is to use a divide and conquer approach. In this algorithm (based on the one published by Bethencourt et. al.[2]), the possible search space is partitioned at each iteration. A partition can be discarded if it contains no sensors, or the partition is of size one, meaning that the location of the sensor has been discovered.

```
Make the set of non-empty partitions={all the search space}.
while the set of non-empty partitions is not empty do
  Extract one of the element from the set of the non-empty partitions. Name it x.
  Partition x up to N partitions.
  For each of the subpartitions of x do:
   Estimate the number of elements in it.
   if the number of elements in the subpartition is zero
      then discard the subpartition.
      else if size of subpartition is equal to one
              then sensor has been located
      else     insert the subpartition into the set of non-empty partitions
```

The maximum running time of the algorithm is O((log U)*S*K). Where: U is the size of the search space; S is the number of sensors; and K in the number of iterations needed to estimate the number of sensors in a partition. The expected running time assuming a uniform distribution of the sensors is also O((log U)*S*K). The change from a linear U dependency to a logarithmic U dependency is due to the comparison in step 7. Once a portion of the search space has been determined without interest, it can be safely disregarded. Thus most of the research has evolved on making this comparison to zero unreliable[2]. The side effect of this algorithm's reduction in time is an increase in resources needed. In particular, the cost of each iteration is the partition size times P. As the maximum partition size is U/N, where N is the maximum number of partitions, the cost per iteration is bounded by U/N*P.

The total cost is then: O(log U * S *K) * cost_per_iteration <= O(log U * S *K * P * U/ N).

3. A RISK BASED APPROACH

The previous analysis assumed is possible to detect a specialized signal injected into the system, by injecting some special packets. While there is no proof that this can be done with 100% certainty, it can be proved that retrieving a signal over time can be done with arbitrary precision given some very lax conditions (This proof is on the appendix). Given this fact, data aggregator designers must optimize the expense, not the possibility of an attack. Like a work factor in cryptography, solutions must have very large bounds. In our analysis, we have assumed that the sensor location is fixed for the duration of the sensor attack. This assumption approximates current practices and limits the usage of the equations, but provides useful guidance for future deployments. This is also the worst case scenario.

The previous equations show dependencies on:

- U: the size of the potential sensor identifications, ie. the a priory size of the anonymity set;
- S: the number of sensors in the aggregation service (S);
- K: the number of iterations required to make a decision, or the number of iterations required to reliably detect the attacker's signal.
- P. the number of packets required per iteration to generate a readable signal.
- N (in the N-ary case), the number of partitions that we can make per iteration or the number of orthogonal signals that we can inject into the system (with the assumption that the costs are the same).

Only two parameters can be controlled by the aggregator service: K and P. The design goal for data aggregator is to implement aggregation methodologies that increase these two values for the attacker while having a smaller effect on the overall aggregated data (This is in lieu of database perturbation methods). Again the key is to measure how well each possible implementation affects both the attacker and the defender.

3.1.1 P: Noise and Sensitivity

The P parameter is the minimum amount of effort required to insert a detectable signal in the published data. This value is directly related to the sensitivity of the system and the noise level of the system. For a linear system (such as the D-shield), P needs to be chosen depending on the average value and the deviation of the undisturbed output. P is also related to the resolution of the output channel, the set of possible output values for each value in the dataset. In general, increasing the size of P reduces the sensitivity of the output or increases the signal to noise ratio.

P can also be thought as an economic disincentive value. Increasing P increases the marginal costs for attackers as more resources are required to extract the identity of any sensor. The precise value and effect on attackers depends not only on P, but also on the problem specific costs per probe. In the case of simple network probes, this cost is almost negligible given the possibility of large botnets[21]. For other types of monitors where more interaction is required, this approach might yield the best results.

Another advantage of P is that it is easy for the aggregator to calculate. The other parameter, K, is harder to estimate, thus, assumptions about its efficacy must be carefully detailed by both designers and deployers of Internet sensors.

3.1.2 K: Uncertainly and Entropy

The K parameter represents a measure of the amount of information that can be extracted from the published data per each interaction. K is an information theoric limit on the properties of the published data which depends on the interaction of the aggregation service with the sensors. In particular, K for the n-ary algorithm is the number of iterations needed to determine estimate with arbitrary precision that there are no sensors present in a subset. Augmenting the K parameter does not imply an increased cost in resources for the attacker but an increased cost in time. An increase in K requires a longer running time that cannot be compensated by more resources (compromised systems). For low interaction systems, where the number of sensors is sufficiently large, the immediate way to generate an increased K is the use of sensor sampling at the aggregator level. For systems which provide richer and more sensitive data, there is no clear way to achieve anonymization while preserving the probabilistic properties of the data. As there is more entropy in the data and this a large place for attackes to put unique 'tokens' in the data. Using sampling at the sensor level, the number of iterations required to determine the presence of a sensor with precision r when the per sampling rate is p is given by: $\log(1-r)/\log(1-p)$. Notice that this value is independent on the how the markings are done or the independent cost per probe.

Another possible way to increase the cost is not to directly increase P or K, but to increase the communication effort needed to potentially scan a host. If sufficient communication overhead is placed on the attacker then the "free" bandwidth and cycles of the compromised machines stops being "free". However this is beyond the control of the aggregator.

It is important to emphasize that it is impossible to prevent the use of the system output as a verification oracle. The goal of the techniques and methods proposed here is to significantly increase the cost of using the system as a verification oracle for multiple systems simultaneously. Confirmation attacks are still possible, but the use of the attacks to explore the address space is no longer feasible.

3.2 An Example with Dshield

Previously discussed is the need to increase the values of P and K as much as possible to make the cost or the time required for an attacker to be sufficiently large. In this section we will discuss mechanisms for a well documented and understood aggregator service: Dshield.

3.2.1 Dshield Operation

Dshield collects data about unexpected connection attempts to computers. Its sensors are end hosts' firewall logs. These logs are given voluntarily to Dshield by the internet community. Dshield aggregates such logs and reports the number of connection attempts per port every hour. Dshield also reports the number of hosts and the number of sensors that observed such behavior. Dshield was the first aggregation service studied for probe attackers by Bethencourt et al.[2] . That work described two types of defenses against probe attacks: social and technical. Social methods include pricing the published data and the use of private reports. But pricing the data would make the data less useful and the use of private reports can only help if there no attackers are also sensors.

Technical measures suggested include: per packet sampling, use of top lists , scan prevention and Delayed reporting. However all of these methods have inherent problems. Per packet sampling generates an increase in P, but does not address the parallelization of the attack. Top lists changes the nature of the reported data. Scan prevention such as the use of IPv6 address space would make the system not useful. Delayed reporting is problematic as late data is of no good for most uses and in fact probe attack efficiency is reduced by only a constant.

We believe that other methods can be more effective at providing the same level of protection to the sensors. In particular the increase in K is not discussed and might be one of the most powerful incentives to prevent such attacks.

3.2.2 Increasing P

The easiest way to increase P is to use of per packet sampling. By selecting a packet to be reported with probability p the attacker must select its reliability measure r (probability of not detecting a sensor) and then he/she needs to send at least $\log(1-r)/\log(1-p)$ packets per iteration per destination.

There are two problems with this approach: First while sampling augments the amount of packets required to detect the signal, it also reduces the noise level and thus some channels that where previously unusable due to noise become available. Second this sampling technique does not over count the packets. Therefore an attacker can use this information to determine an upper bound on sensors in a partition. The attacker can end probing on a subpartition when that bound is reached.

A potentially better method is the use of randomized sampling. At each period each sensor selects a probability between p1 and p2 (with uniform distribution between these two values). By using this method three things happen. First, the attacker must use the lowest probability to guarantee that his signal is observed while the sensors average probability is (p1+p2)/2. Second, this introduces some noise factor. Third, probes can be over counted, this overcounting prevents the attacker from discarding any sub partitions when thresholds are reached. The advantage is the information asymmetry of the method. There is a difference between the guaranteed probability of selection and the expected probability of selection. In other words, this method disturbs the data more effectively for attackers than for defenders.

Other approaches include the use of buckets of defined sizes to group data or limiting the resolution of the output signal. Resolution is deceased by limiting the number of significant digits of the output. However the effect of these methods is similar to the simple per packet sampling.

3.2.3 Increasing K

This section provides multiple mechanisms to increase K. Recall K is the information theoric limit on the cost function. One way to potentially increase K is also to use sampling, but at the sensor level. At each time interval, the aggregation service will select the logs from some sensor to be added to the aggregate list with some probability p. By using this sampling, an attacker signal for each sensor would be lost at each interval with probability (1-p) no matter what type of signal he/she introduced.

Another way to increase K is the use of data correlations. As Dshield is designed to detect automated threats, we can use certain domain specific knowledge about such threats. Data are uncorrelated in the source IP address. Data are also uncorrelated in the time domain. With this in mind, assume that the lower X bits of IP address space are uniformly distributed and use them to sample. By sampling on one of these bits, an attacker using only one compromised machine has a 50% change of not being reported, independent of the number of probes sent to the sensor. Time is the other possible correlation dimension we can use . The time sampling mechanism could select randomly only even seconds or only on the first half hour (or every uneven packet). This would force the attacker to not only use more resources but also to spread them in a more uniform distribution. This requires synchronization among all the systems used by the attacker to launch the attack.

In general, the use of data correlations does not directly increase K, but provides a large disincentive to try to determine the location of a sensor. The probability of observing the output, given an attacker that can generate different packets that match each of our selection dimensions is given by: $p_{randomselection}+(n-1)/N$

Where

- $P_{andomselection}$ is the base probability of selecting any one random packet ,
- n is the number of probes sent by the attacker that are in different dimension
- N is the total number of possible selections.

Another option is to use a Markov chain to select whether a sensor reports data back to the aggregate sensor. While this does not provide extra protection it increaes the complexity of the attack. The order of the running time remain the same, but the attacker is forced to store more state. Specifically attackers must interleave the sampling of data and cannot use depth first attacks. Markov chains and data correlations are examples of the use of information asymmetry.

The requirement of state date makes the attack more expensive while not changing the accuracy of the collected data.

3.2.4 Limitations of the methods

All methods discussed in the increase of K and P have two potential problems. The first is that the data quality of the collection system is decreased. However, assuming the distributions are uniform in the dimensions selected for sampling, adding noise does not change the expectation of the output before and after using the proposed methods. The real problem comes from calculating the expected deviation of this output given each sampling mechanism.

The second potential pitfall is the that the system is more sensitive to rogue sensors. The effects of rogue sensors can be amplified with sampling is the sensor implements the sampling and provide malicious data. However, if the sensors are required to submit sampled sanitized data then abnormal deviations of sampling values can be detected. The sensor system still needs to use other tools to validate the data reported by individual sensors, but this question is out of the scope of this paper.Please use a 9-point Times Roman font, or other Roman font with serifs, as close as possible in appearance to Times Roman in which these guidelines have been set. The goal is to have a 9-point text, as you see here. Please use sans-serif or non-proportional fonts only for special purposes, such as distinguishing source code text. If Times Roman is not available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged.

4.CONCLUSIONS

Anonymization procedures employed by aggregation services are a unique and important special cases of anonymity. Without proper anonymization, those services vulnerable to injection attacks and reduces the confidence of sensors. The absence of an absolute method to assure anonymity for sensors indicates that economic and information theoric approaches are needed. We have shown the fastest known algorithm for sensor location and the kinds of mitigation mechanisms that can be put in place. We introduced two parameters that can be used to explain the effectiveness of potential mitigation solutions.

In the particular case of the Dshield we have enumerated the problems of other currently proposed defense mechanisms. We have offered a set of methods that can be used to avoid said enumerated problems such as the use of randomized packet sampling, sensor sampling and correlation sampling. Further we have shown that specificity when describing sampling methodologies is required. Sampling in different spaces generates different dependencies.

Further research is required into the efficacy of leveraging determining information asymmetries. Currently we are working to determine how robust our proposed methods are against rogue sensors. Currently deployed data aggregators must implement defense mechanisms as soon as possible in order to guarantee the accuracy of their data set. Future aggregation services must spend more time in the analysis of the anonymization mechanisms specifically in on how to generate anonymization methods with highest marginal costs for attackers.

5.ACKNOWLEDGMENTS

This work is sponsored by Indiana University's Advanced Network Management Lab (ANML) and the Institute for Information Infrastructure Protection (I3P) research program..

6.REFERENCES

[1] Rakesh Agrawal and Ramakrishman Srikant. Privacy-preserving data mining. In ACN SIGMOD 2005, May 2000.

[2] John Bethencourt, Jason Franklin, and Mary Vernon. Mapping internet sensors with probe response attacks. In Proceedings of the 14th USENIX Security Symposium, August 2005.

[3] Hossein Bidgoli, editor. The Handbook of Information Security, chapter Cryptographic Privcy Protection Techniques. John Wiley and Sons, Inc., 2005.

[4] Joachim Biskup and Ulrich Flegel. On pseudonomization of audit data for intrusion detection. In Workshop on Design Issues in Anonymity and Unobservability, July 2000.

[5] Jinliang Fan, Jun Xu, Mostafa H. Ammar, and Sue B. Moon. Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-based sheme. Comput. Networks, 46(2):253–272, 2004.

[6] Ulrich Flegel. Pseudonymzing unix log files. In Proceedings of the Infrastructure Security Conference, October 2002.

[7] --,"CAIDA Cooperative Association for Internet Data Analysis". CAIDA website (http://www.caida.org), July 2005.

[8] Lake V.S. Lakshmanan, Raymond T. Ng, and Ganesh Ramesh. To do or not to do: The dilemma of disclosing anonymized data. In SIGMOD, pages 61–72, June 2005.

[9] Greg Minshall. Tcpdpriv man page. http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html, 1997.

[10] The myNetWatchman Project. The mynetwatchman website. http://www.mynetwatchman.com.

[11] Ruoming Pand and Vern Paxson. A highlevel programming environment for packet trace anonymization and transformation. In SIGCOMM' 03 . ACM, August 2003.

[12] Charles Simpson. Neti@home website. www.neti.gatech.edu, August 2005.

[13] Latanaya Sweeney. Computational Disclosure Control: Aprimer on Data Privacy Protection.PhD thesis, Massachusetts Institute of Technology, June 2001.

[14] Latanaya Sweeney. k-anonymity: A model for protecting privacy. International Journal on Uncertainty Fuzziness and Knowledge-based Systems , 10:555–570, 2002.

[15] Michael E. Locasto, Janak J. Parekh, Angelos D. Keromytis, Salvatore J. Stolfo, Towards Collaborative Security and P2P Intrusion Detection. In Proceedings of the 2005 Information Assurance Workshop June 2005.

[16] Johannes Ullrich and Euclidian Consulting. Dshield homepage. www.dhield.org, August 2005.

[17] University of Michigan, Merit Network and Arbor Networks. University of Michigan Internet Motion Sensor. http://ims.eecs.umich.edu 2005.

[18] Esther Gal-Or and Anindya Ghose. The Economic Consequences of Sharing Security Information. Workshop on the economics of information security 2003.

[19] The Honeynet Research Alliance. http://www.honeynet.org/alliance/index.html last visited July 2006.

[20] REN-ISAC monitoring website. http://www.ren-isac.net/monitoring.html last visited July 2006.

[21] The Honeynet Project. Know your enemy: Tracking Botnets. http://www.honeynet.org/papers/bots/ .2005.

[22] Andrei Serjantov and George Danezis. Towards an Information theoric Metric for anonymity. In proceedings of the 2002 privacy enhancing technologies workshop (PET02). 2002.

[23] Patrick Lincoln, Philip Porras and Vitaly Shmatikov. Privacy Sharing and Correlation of Security Alerts. USENIX security 2004.

[24] Department of Homeland Security. PREDICT system. www.predict.org. Last accessed September 2006.

[25] Jianqing Zhang, Nikita Borisov, and William Yurcik, " Outsourcing Security Analysis with Anonymized Logs ," *2nd IEEE SECURECOMM SECOVAL Workshop*, Baltimore MD. USA, September , 2006.

[26] Justin Brickell and Vitaly Shmantikov. Efficient Anonymity-Preserving Data Collection. KDD 2006.

[27] Roger Dingledine, Nick Mathewson, Paul Syverson. Tor: The Second-Generation Onion Router. Usenix Security 2004, August 2004.

[28] Richard Clayton, George Danezis and Markus G. Kuhn. Real World Patterns of Failure in Anonymity Systems. Information Hiding 2001.

[29] Evan Cooke, Michael Bailey, Morley Mao and Danny McPherson. Toward Understanding Distributed Blackhole Placement. Worm '04 October 2004.

APPENDIX

Sensor identification in discrete spaces:

What we have then is a problem of system identification. These type of problems are very common in the control theory, and our problem tough similar hast three properties that make them a little bit different: (i) exact knowledge of complexity of the transfer function, (ii) large input space, and (iii) non-zero mean (or median) error. The exact knowledge of the complexity of the transfer function means that we know the exact structure of the system we want to identify, thus the identification task is to generate good approximations for the parameters of that structure. The fact that this structure is known a priori in general reduces the complexity of the identification procedure. Usually identification systems have a relatively small input space of order less than 10^3 where in our case we have a large set of inputs, that is all valid Internet end points around10^9 . This means that methods that require large number of input-output probes cannot be used as the space cannot be generated or stored. The non-zero mean error means that some techniques as sum of squares cannot be used directly.

However, the theory and knowledge of system identification procedures can still be used but with some caveats. For our case the critical part is to determine the excitation signals necessary for appropriate identification. These signals must satisfy two conditions: They must cover as much as possible the internal state of the system and they must be detectable(identifiable) in the output. Since an attacker can reach any end point in the system, what really requires study is the detection of the signal.

A.1Signal detection in discrete spaces

Discrete signal detection is a known problem in communication systems. In particular, in cases where the transmission channel is linear and the noise is with finite energy and with zero mean (ex. white noise) methods to detect signal are pretty much known. Our case in particular has a finite output space and the signal is also discrete in time. Our function is not linear (in general) and the noise is bounded and has has non-zero mean. But even in these case, signal detection with an arbitrary non-zero error is possible under the following circumstances:

1.Ability to excite the channel

2. Noise is i.i.d. (Independent identically distributed) at each time period.

3. System is time invariant.

4. The conditional PDF of the output in the case with no signal is known.

5. For at least one of the possible output values y_i the conditional probability, given the signal is present is known to differ from the no signal case by at least some known ε_i. In other words: $\exists y_i / \| P(y_i | Nosignal) - P(y_i | Signal) \| \geq \epsilon_i$

A.2 Proof, binary Case

If the output function is binary, from condition (4) we know the signal less distribution d 1 with parameters: p1 and q1 = 1 − p1 . Since this distribution is time invariant (conditions (2), (3)) , a sequence of outputs of this distribution will generate a binomial distribution. This binomial distribution with N trials has:

$$\begin{aligned} \mu_{d1} &= Np_1 \\ \mu_{d1,2} = \sigma^2_{d1} &= Np_1 q_1 \end{aligned} \quad \text{(a.1)}$$

From the conditions for identification(condition (5)) we know that for the signal case we have p2 p1 + (Or the opposite, in which we can rename the output signals). In this case (with signal) the repeated trial would yield to another binomial distribution with:

$$\begin{aligned} \mu_{d2} &= Np_2 \\ \mu_{d2,2} = \sigma^2_{d2} &= Np_2 q_2 \end{aligned} \quad \text{(a.2)}$$

Now, we want the error to be least that some p_{req}. If we set the decision threshold in the midpoint between the two expectations t = (Np1 + Np2)/2. The error of detection is given by the maximum area where the decision threshold gives the opposite value. What is needed is to find an N which the error would be less than that. Using the Tchevycheff's inequality we can say:

$$P|X_{d1}-\mu_{d1}\geq t|\leq\frac{\sigma_{d1}^{2}}{t^{2}}\leq p_{req} \quad \text{(a.3)}$$

Replacing in equation (4) with values from equation

[30] we can come with:

$$\frac{Np_1q_1}{(N(p_2-p_1)/2)^2}\leq p_{req}$$

$$\frac{4p_1q_1}{N(p_2-p_1)}\leq p_{req}$$

Thus:

$$N\geq\frac{p_1q_1}{p_{req}(p_2-p_1)^2}$$

Similarly for the second distribution (with signal) we have:

$$N\geq\frac{p_2q_2}{p_{req}(p_2-p_1)^2}$$

Thus we can find a bound for N that satisfies the requirement for an arbitrary but non-zero error requirement p_{req} .

A.3 Arbitrary Case

We can convert an arbitrary function that we know at least some □$_i$ into the binary case. And we can use the above proof.