**Edith Cowan University**
**Research Online**

Australian Digital Forensics Conference | Conferences, Symposia and Campus Events

2007

# Extracting inter-arrival time based behaviour from honeypot traffic using cliques

Saleh Almotairi
*Queensland University of Technology*

Andrew Clark
*Queensland University of Technology*

Marc Dacier
*Institut Eurécom*

Corrado Leita
*Institut Eurécom*

George Mohay
*Queensland University of Technology*

***See next page for additional authors***

DOI: 10.4225/75/57ad42dd7ff2c
Originally published in the Proceedings of the 5th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2007.
This Conference Proceeding is posted at Research Online.
http://ro.ecu.edu.au/adf/8

**Authors**
Saleh Almotairi, Andrew Clark, Marc Dacier, Corrado Leita, George Mohay, Van Hau Pham, Olivier Thonnard, and Jacob Zimmermann

This conference proceeding is available at Research Online: http://ro.ecu.edu.au/adf/8

# Extracting Inter-arrival Time Based Behaviour from Honeypot Traffic using Cliques

Saleh Almotairi[1], Andrew Clark[1], Marc Dacier[2], Corrado Leita[2],
George Mohay[1], Van Hau Pham[2], Olivier Thonnard[2], Jacob Zimmermann[1]
[1]Queensland University of Technology, GPO Box 2434, Brisbane 4001, Australia
[2]Institut Eurécom BP 193, F06904 Sophia Antipolis cedex, France
{s.almotairi, a.clark, g.mohay, j.zimmerm}@isi.qut.edu.au
{marc.dacier, corrado.leita, van-hau.pham}@eurecom.fr olivier.thonnard@rma.ac.be

## Abstract

*The Leurre.com project is a worldwide network of honeypot environments that collect traces of malicious Internet traffic every day. Clustering techniques have been utilized to categorize and classify honeypot activities based on several traffic features. While such clusters of traffic provide useful information about different activities that are happening in the Internet, a new correlation approach is needed to automate the discovery of refined types of activities that share common features. This paper proposes the use of packet inter-arrival time (IAT) as a main feature in grouping clusters that exhibit commonalities in their IAT distributions. Our approach utilizes the cliquing algorithm for the automatic discovery of cliques of clusters. We demonstrate the usefulness of our methodology by providing several examples of IAT cliques and a discussion of the types of activity they represent. We also give some insight into the causes of these activities. In addition, we address the limitation of our approach, through the manual extraction of what we term supercliques, and discuss ideas for further improvement.*

**Keywords**

Honeypots, Internet traffic analysis, clustering, inter-arrival times

## INTRODUCTION

The work described in this paper builds upon previous work by Pouget et al. (2006) in the use of data obtained in the *Leurre.com* environment for detecting anomalous Internet traffic. This distributed low interaction honeypot environment currently consists of 50 platforms located in 30 different countries. In the previous work we have shown that the analysis of the inter arrival times (IATs) between packets collected in this environment could provide a valuable contribution to network forensics.

In that work, we used the notion of *attack clusters* proposed in earlier work by some of the same authors and we introduced the notion of cliques of clusters as an automated knowledge discovery method. A clique is a group of clusters that share common characteristics related to one or maybe a few attack processes.This paper revisits the approach to identifying IAT-based cliques in an attempt to achieve the automatic derivation of cliques of clusters with common IAT characteristics to better identify the repeated use of commonly used tools, and also to identify spurts of activity by such commonly used tools. We present a systematic approach for building new cliques and provide an extensive validation of the approach over large datasets. Last but not least, by means of *classes of cliques*, we show the usefulness of the approach as a result of the new knowledge they help to derive about the attack traces collected.

The structure of the paper is as follows. First we define our old notion of clusters as well as our new definition of the cliques and the process to build them. We then validate the approach thanks to experimental results carried out over data obtained during a period of 3 months (March to May 2007) in the Leurre.com environment. Finally we provide conclusions and discussion about the results of the paper.

## CLUSTERS AND CLIQUES

In this section we describe the techniques used to classify the traffic collected by the various honeypots. We first introduce the basic concepts of the *Leurre.com* traffic analysis. We then recall the *Leurre.com* clustering algorithm, and then present the novel cliquing algorithm introduced in this paper.

### Terminology

A *platform* is one of the many *Leurre.com* honeypot sites. Each platform contains three *virtual hosts*, with distinct IP addresses, impersonating three different OS behaviours taking advantage of the *honeyd* software. The

honeypots are configured to obtain a minimal level of interaction, replying to ICMP echo requests and establishing TCP connections on their open ports.

All activity observed by the honeypots is attributed to a *source,* meant to uniquely identify an attacker taking into consideration dynamic addressing. Two activities generated by a given IP address and separated by a period of more than 25 hours are attributed to two different sources.

A *large session* is a collection of packets exchanged between one source and one platform, while a *tiny session* groups the packets exchanged between one source and one virtual host. A large session is thus composed of up to three tiny sessions, ordered according to the virtual hosts IP addresses.

A *port sequence* is the sequence of (TCP or UDP) ports targeted on a virtual host, within a tiny session.

A packet *inter-arrival time* or *IAT*, is the time difference (in seconds) between the arrival of two consecutive packets at a virtual host (i.e., within a tiny session).

**Clustering Algorithm**

The first step of the clustering algorithm consists in grouping large sessions into *bags*. This grouping aims at differentiating between various classes of activity taking into consideration a set of preliminary discriminators, namely the number of targeted virtual hosts and the *unsorted* list of port sequences hitting them.

In order to further refine the bags, a set of continuous parameters is taken into consideration for each large session, namely: its duration, the total number of packets, the average IAT, and the number of packets per tiny session. These parameters can assume any value in the range $[0, \infty]$, but some ranges of their values may be used to define bag subclasses. This is done through a peak picking algorithm that identifies ranges of values considered discriminating for the bag refinement. Large sessions belonging to a bag and sharing the same matching intervals are grouped together in a *cluster*.

A very last refinement step is the *payload validation*. The algorithm considers the concatenation of all the payloads sent by the attacker within a large session ordered according to the arrival time. If it identifies within a cluster multiple groups of large sessions sharing similar payloads, it further refines the cluster according to these groups.

**Cliquing Algorithm**

Due to the large quantity of data we collect, we need to rely on an automated methodology that is able to extract relevant information about the attack processes. Our correlative analysis relies on concepts from graph and matrix theory. In this context, a *clique* (also called a complete graph) is an induced subgraph of an (un)directed graph in which the vertices are fully connected. In our case, each node represents a cluster, while an edge between a pair of nodes represents a similarity measure between two clusters. The main focus of this work is on computing similarities between IAT distributions, but our methodology can be applied to any type of vector or time series.

Determining the largest clique in a graph is often called the *maximal clique problem* and it is a classical graph-theoretical, NP-complete problem (Bron and Kerbosch, 1973). Although numerous exact algorithms (Kumlander 2004a, 2004b, Bomze et al. 1999) and approximate methods (Bomze et al. 2000, Pavan and Pelillo 2003) have been proposed to solve this problem, we address the computational complexity of the clique problem by applying our own heuristics to generate sets of cliques very efficiently. While our technique is relatively straightforward, it possesses two significant features. Firstly, our technique is able to deliver very coherent results with respect to the analysed similarities. Secondly, regarding the computational speed, our technique outperforms other algorithms by several orders of magnitude. For example, we applied the approximate method proposed by Pavan and Pelillo (2003) which consists of iteratively extracting *dominant sets* of maximally similar nodes from a similarity matrix. On our dataset, the total computation was very expensive (several hours) whereas our custom cliquing algorithm only takes a few minutes to generate the same cliques of clusters with the same dataset.

On the other hand, our heuristic imposes a constraint on the similarity measure, namely that it has to be *transitive.* With this restriction, it is sufficient to compute the correlation between one specific node and all other nodes in order to find a maximal clique of similar nodes. We achieve this transitive property by carefully setting a global threshold on the measurement of similarities between clusters (see next section).

Here are the different steps of our cliquing algorithm:

1. We define a quantitative representation for the feature to correlate (in this work: the IAT distribution within clusters).

2. We choose a well-suited similarity measure for this characteristic.
3. Consider the list of all clusters. While this list is not empty:
    - We consider the next cluster in the list and we take the corresponding characteristic vector;
    - We compute the similarities with all other remaining vectors;
    - If there are other similar clusters (with respect to the defined threshold), we put all of them in a new clique. We remove those clusters from the list and start the next iteration.
    - If there is no other similar cluster, we remove the current cluster from the list, store it in a separate group, and start the next iteration.

Clearly, this algorithm takes advantage of the already created cliques to progressively decrease the search space; so in the average case the algorithmic complexity will be less than $O(n^2)$, and we could expect typically a complexity order of $O(n.log(n))$. The exact complexity analysis of our algorithm is out of the scope of this paper.

**Cluster Correlation using Packet Inter-arrival Times**

The first step in our methodology is to construct the cluster characteristics. We represent the IAT distributions of the clusters with a vector in which every element corresponds to the IAT frequency of a pre-defined bin (range of time values). We end up with an IAT vector of 152 bins where the first bin groups IATs falling in the interval 0-3 seconds, and the last bin corresponds to IATs of 25 hours or more.

To circumvent the limitations of our previous work, we now rely on a similarity measure that is based on a recent technique called *symbolic aggregate approximation* (SAX) (Lin et al. 2003). SAX aims at reducing a complex time series to a symbolic approximation without losing too much quality with respect to the shape of the signals. It is a *piecewise aggregation approximation* (PAA) technique which tends to approximate time series by segmenting them into intervals of equal size and summarizing each of these intervals by its mean value.

SAX uses predetermined breakpoints during the quantization, chosen so as to maximize the energy of the quantized representation of the time series, which are then interpreted as a string of symbols taken from a finite alphabet. Figure 1 gives an example of a time series converted to a SAX representation which has been mapped to the string *eefeffeccbbabaab* . A SAX representation of a time series T of length N can be denoted by $W_T$ (N,w,$\alpha$), where: *N* is the number of elements in T; *w* is the number of elements in the SAX representation of T (i.e. the length of $W_T$); and $\alpha$ is the alphabet size (number of quantization levels). The ratio $r = N/w$ is called the compression ratio. A value of r = 10 means that 10 elements of T are mapped to a single symbol in $W_T$.

One of the strong advantages of SAX resides in the fact that this technique allows a distance measure that lower bounds the original distance measure (e.g. the Euclidean distance, see Lin et al. (2003) for the proof). SAX defines a MINDIST function that returns the minimum distance between the original time series of two words. Let $T_1$ and $T_2$ be two time series of same length N, then the minimum distance given by SAX can be calculated as follows:

$$MINDIST\left(W_{T_1}, W_{T_2}\right) = \sqrt{\frac{N}{w}} \cdot \sqrt{\left(\sum_{i=1}^{w} dist\left(W_{T_1}(i), W_{T_2}(i)\right)\right)^2}$$

The *dist()* function returns the inter-symbol distance and can be implemented using a table lookup for better computational efficiency (see Lin et al. (2003) for more details).

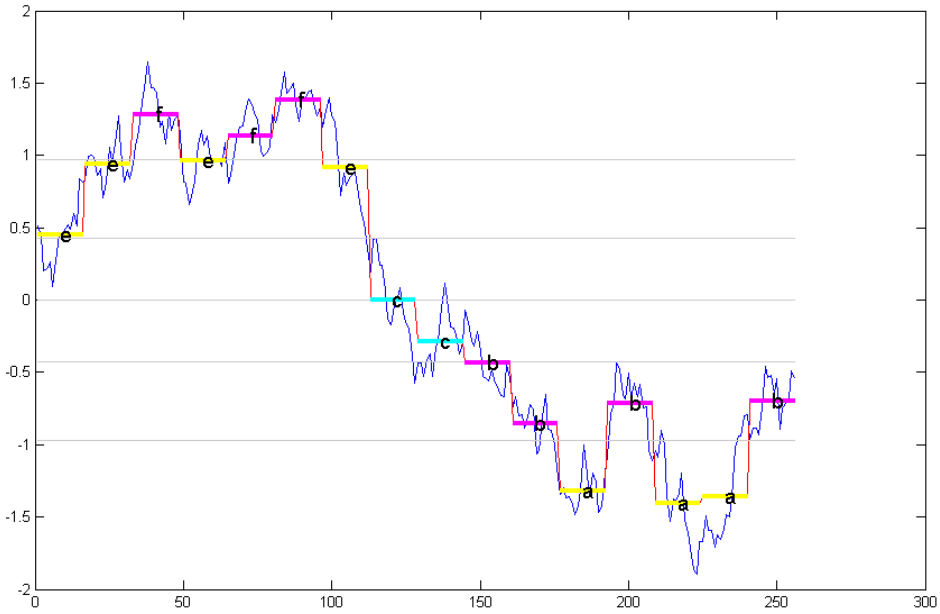


*Figure 1. Example of SAX representation of a time series $W_T(256, 16, 6)$.*

In order to achieve a transitive similarity function, we set a global threshold on the distance computed with SAX. Only if the similarity measure exceeds 99% of the maximal theoretical value, do we assume that the two vectors are completely similar. This experimental heuristic gives fair results. A drawback of this approach, as for every method which relies upon a threshold, is that a good preliminary tuning is needed to fit it to the data.

SAX can also typically compress the time series, but we chose here not to compress the IAT vectors because we already defined packet IAT bins which regroup all values falling in those respective intervals.

## EXPERIMENTAL RESULTS

We now describe our analysis of the IAT-based cliques obtained using the above approach when applied to the *Leurre.com* dataset. We consider a dataset covering three months of traffic (March May 2007) collected from the *Leurre.com* environment.

For the sake of conciseness, in the analysis presented in this paper, we only consider clusters which have at least one bin, after the 21st bin (the 22nd bin corresponds to around five minutes), with a count of more than 10. This means that we ignore clusters which do not have more than 10 occurrences of at least one IAT value greater than five minutes.

After this filtering, we obtained 1475 vectors representing the IAT frequency distributions of the corresponding clusters. The clique algorithm described above was then applied to these vectors, yielding 111 IAT-based cliques comprising 875 clusters. The remaining 600 clusters did not fall into any clique.

Each clique contains a group of clusters which, based upon their IAT distribution (and the parameters of the cliquing algorithm) are similar. Prior to describing our detailed analysis of the cliques obtained we present three types of cliques that we expected would, *inter alia*, be represented in the results:

**Type I:** Cliques which contain clusters of large sessions targeting the same port sequences. The difference between the various clusters contained within such a clique lies in the number of packets sent to the targeted ports. These cliques are mostly symptomatic of classes of attacks where the attacker repeatedly tries a given attack, a varying number of times.

**Type II:** Cliques composed of clusters of large sessions targeting different port sequences but exhibiting the same IAT profile. These cliques are symptomatic of tools that send packets to their target according to a very specific timing and that have been used in several distinct campaigns targeting different ports.

**Type III:** Cliques which contain clusters grouped together based upon the presence of long IATs (longer than 25 hours), representing sources which are observed on one platform, then, within 25 hours, are detected on another platform, before again returning to the original platform. Such behaviour

would be indicative of a source which is scanning large numbers of devices across the internet, in a predictable manner, resulting in them repeatedly returning to the same platform.

We also found many similarities across the different cliques that were generated. We identified a number of so-called *supercliques* as a result which suggests that the IAT-based analysis we have focused on in this paper is good at automatically identifying very specific types of activity within a very large dataset. Our analysis of these supercliques is presented below.

**Type I Cliques**

Type I cliques are expected to contain clusters which are very similar with respect to most traffic features, including port sequence, with one exception being that the large and tiny sessions within the clusters contain varying durations (both in terms of time, and the number of packets sent by the source). The variation in the duration of the sessions will account for such traffic being arranged in different clusters. Two particular cliques that are seen to fall clearly into the Type I category are Clique 7 and Clique 49 (summarised in Table 1).

Clique 7 is composed of 8 clusters, 9 large sessions and a total of 821 packets. These clusters are mainly contained in one bag (corresponding to the very common port sequence of TCP/135). In this clique, there are 5 platforms targeted by 6 distinct IP addresses originating from 4 different countries (China, Germany, Japan, and France). The peak IAT bin is bin 32 with IAT values in the range 554-583 seconds, and the average duration is 70491 seconds with a minimum duration of 4657 seconds, and a maximum of 236350 seconds.

All three virtual hosts on each of the targeted platforms were hit with the same number of packets, with the average number of packets per session equal to 35. Also, several IP addresses were found to occur in multiple clusters within the clique. While these sources were grouped in different clusters due to their varying durations, there were strong similarities in terms of the IAT characteristics of the sessions, resulting in these clusters being grouped in the same clique.

| Clique | Clusters | Large Sessions | Packets | Bags | Platforms Targeted | Source IPs | Countries | Targeted port sequence | Peak IATs (bin) | Min, average, max durations in secs | hosts per platformNo of target virtual |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 8 | 9 | 821 | 1 | 5 | 6 | 4 | TCP/135 | 554-583 (32) | 4657, 70491, 236350 | 3 |
| 49 | 11 | 285 | 3274 | 1 | 37 | 248 | 46 | TCP/22 | 2703-3597 (49) | 1035, 9922, 137528 | 3 |

*Table 1: Type I Cliques*

Clique 49 contains 11 clusters, 285 large sessions, and 3274 packets, and the targeted port sequence is TCP/22. There are 248 distinct IP addresses which attacked 37 different platforms. The sources of the IPs are widely spread among 46 different countries. Despite the widespread location of the sources of the traffic in this clique, there are a number of similarities in the observed behaviour. Firstly, large sessions in this clique always targeted all three virtual hosts on each platform, and the number of packets sent to each virtual host was similar in each case (one packet for the Windows hosts and an average of 10 packets for the UNIX host). The average duration of attacks is 9922 seconds with minimum and maximum durations in the range of 1035 to 137528 seconds. The majority of the clusters in this clique belong to the same bag. The IAT sequences of these clusters are similar with all IATs in the session being short except one which belongs to bin 49 (2703-3597 seconds).

Cliques 7 and 49 were typical examples of Type I cliques where attack traffic ends up in different clusters due to the variations in either the duration of the attack or the number of packets sent. In each case the duration and number of packets varied significantly between the sessions, while the IAT behaviour remained consistent.

Also, a number of IP address were shared between clusters within each clique, with over 50 % of the clusters sharing IP addresses or class C networks.

The identification of cliques of Type I addresses a weakness of the original clustering algorithm which was, by design, unable to group together activities that clearly were related to each other and should have, therefore, be analysed together.

**Type II Cliques**

Type II cliques are those which contain a large variety of targeted port sequences, yet each cluster exhibits similar IAT characteristics. We hypothesise that clusters belonging to this type of clique correspond to the same attack tool using the same strategy to probe a variety of ports (such as a worm which targets multiple vulnerable services, or some other type of systematic scanner targeting a number of different ports). Two cliques which exhibit this type of behaviour are Cliques 92 and 69 (see Table 2).

| Clique | Clusters | Large Sessions | Packets | Bags | Platforms Targeted | Source IPs | Countries | Targeted Port Sequences | Peak IATs (bin) | durations in secsMin, average, max | platformNo of target virtual hosts per |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 92 | 40 | 502 | 4234 | 7 | 1 | 502 | 25 | (all TCP)<br>6769<br>7690<br>12293<br>18462<br>29188<br>64697<br>64783 | 933-1797 (46);<br>1803-2702 (48); | 953,<br>9278,<br>53941 | 1 |
| 69 | 64 | 1336 | 17097 | 8 | 2 | 1300 | 37 | (all TCP)<br>4662<br>6769<br>7690<br>12293<br>29188<br>38009<br>64697<br>64783 | 933-1797 (46) | 133,<br>44163,<br>22522<br>4 | 1 |

*Table 2: Type II Cliques*

Clique 92 consists of 40 clusters, 502 large sessions and 4234 packets in total. While a variety of ports are targeted by these clusters, traffic within each cluster only targets a single port. The TCP ports targeted within this clique are: 6769, 7690, 12293, 18462, 29188, 64697, and 64783. This clique is a result of 502 distinct source IP addresses originating from 25 different countries, and targeting only a single platform. Additionally, only one virtual host was targeted on this platform. The average number of packets per large session was 16 (minimum 3 and maximum 103), and the average duration was 9278 seconds. Clusters in this clique belong to 7 different bags (corresponding to the 7 different ports targeted). Clique 92 contains peak IAT bins of 46 (933-1797 seconds) and 48 (1803-2702 seconds) where the IAT sequences are repeated patterns of short and long IATs. A possible explanation for the traffic which constitutes this clique is that it corresponds to the same tool being used to scan for the existence of services which use a strange port (such as peer-to-peer related services) where the scan uses a regular (long) delay between retransmissions.

Clique 69 is similar to Clique 92 in that it also contains a variety of clusters where each cluster contains traffic targeting a single, unusual port. This clique contains 64 clusters, 1336 large sessions and 17097 packets. It is a result of 1300 distinct attacking IP addresses, that originate from 37 different countries and target 2 platforms (all but one target the same platform as that targeted by the traffic in Clique 92). The targeted TCP ports are: 4662, 6769, 7690, 12293, 29188, 38009, 64697, and 64783. Clusters in this clique belong to 8 different bags, and in each case only one virtual host was targeted per platform. The durations of attacks range from 133 to 225224 seconds with an average of 44163 seconds. The number of packets sent in each large session is in the range 2 to 135 with an average of 25 packets. The IAT sequences are repeated patterns of short, short, and long IATs with a peak IAT bin of 46 (933-1797 seconds).

The traffic in Cliques 92 and 69 represent a large number of distinct sources from a variety of counties targeting a variety of ports, predominantly (with one cluster being the exception), targeting the same platform in China. These cliques represent very interesting activity which is difficult to characterise in further detail due to the lack of interactivity of the honeypots on these ports. The significance of the ports being targeted is unclear, but might be easier to determine if packet payloads were available. The fact that all of these sources exhibit a very distinct fingerprint in terms of their IAT characteristics makes the activity all the more unusual.

The identification of cliques of Type II enables us to highlight, in a systematic way, the existence of tools with a specific IAT profile that are reused to launch different attack campaigns against various targets. Without such analysis, the link that does exist between the IPs belonging to different clusters in a given clique would have remained hidden.

**Type III Cliques**

Based upon our observation of the *Leurre.com* data over a long period of time, we found that there are a number of large sessions which continue for an extended duration (sometimes many weeks). Of these there are a number which target multiple platforms within a 25 hour period, where the intervening time before returning to the same platform is more than 25 hours. These very long IATs are placed into bin 152 during the cliquing process. A number of cliques that resulted from the cliquing algorithm were characterised by these long IATs, and here we investigate two of them in detail Cliques 31 and 66 (seeTable 3).

Clique 31 is a large clique of 150 clusters, 3456 large sessions, and a total of 21422 packets. The port sequence for Clique 31 is the single port UDP/1434 (MS SQL). In Clique 31, there are 277 distinct IP addresses originating from 22 different countries which target 39 different platforms. Characteristics of clusters in this clique include: a varying number of hosts targeted, with the average number of packets sent per host equal to 12 (minimum 2 and maximum 85) and an average duration equal to 1142131 seconds. Clusters in this clique belong to 7 different bags and have IAT values that exceed 25 hours (IAT peak bin 152). These sessions are indicative of a very slow scanner which is seen on multiple platforms, returning to the same platform only after an extended delay of more than 25 hours.

| Clique | Clusters | Large Sessions | Packets | Bags | Platforms Targeted | Source IPs | Countries | Port s targeted | Peak IATs in secs (bin) | Min, average, max durations in secs | hosts per platformNo of target virtual |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 31 | 150 | 3456 | 21422 | 7 | 39 | 277 | 22 | UDP/1434 | >25 hours (152) | 132, 1142131, 7509849 | varies |
| 66 | 3 | 13 | 171 | 3 | 12 | 9 | 2 | UDP/1026; UDP/1027 | Very large (152) | 1, 381408, 915002 | 3 |

*Table 3: Type III Cliques*

Clique 66 contains 3 clusters, 13 large sessions and 171 packets. These sessions are characterised by sending multiple packets, alternating between UDP ports 1026 and 1027 repeatedly. In Clique 66, 12 platforms were targeted by 9 distinct IP addresses originating from 2 different countries. All clusters within this clique contain sessions which target all three virtual hosts on the target platforms, with only a small number of packets sent per session (on average 4, with a minimum of 3, and a maximum of 6). The average session duration is 381408 seconds. Clusters in this clique belong to 3 different bags and, again, the IAT durations are very large.

Cliques 31 and 66 represent examples of activities where a source IP is scanning the globe, targeting different honeypot platforms in less than 25 hours. UDP port 1434 is used by the MS SQL Monitor service and is the target of several worms, such as W32.SQLExpWorm and Slammer. It is likely that traffic targeting this port is result of worms that scan for vulnerable servers. UDP ports 1026 and 1027 are common targets for Windows Messenger spammers, who have been repeatedly targeting these ports since June 2003[1].

**Supercliques**

We observed that the across all of the obtained cliques, only a relatively small number of peak IAT bin values were represented. Indeed, from the point of view of the peak bin values, we found that a limited number of combinations existed. This suggests that the cliques we obtained possess a high level of uniformity in terms of the activities that they represent. Based upon the small set of common peak bins, and the dominant port sequences targeted within those cliques, we manually grouped the cliques together into 6 *supercliques*, which are summarised in Table 4.

| Superclique | Cliques | Clusters | Large Sessions | Distinct IPs | Peak Bins | Port Sequence |
|---|---|---|---|---|---|---|
| 1 | 7 | 166 | 3505 | 277 | 152 | 1434U |
| 2 | 5 | 12 | 22 | 12 | 152 | 1026U1027U … |
| 3 | 6 | 29 | 288 | 247 | 46, 48, 49 | 135T |
| 4 | 4 | 21 | 541 | 429 | 46, 48, 49 | 22T |
| 5 | 23 | 183 | 6313 | 6188 | 46, 48, 49 | unusual TCP ports |
| 6 | 23 | 74 | 164 | 152 | 31, 32 | 135T |

*Table 4: Supercliques and their representative properties.*

As can be seen from the table, the supercliques account for just over half of the cliques generated. The cliques not represented within the supercliques were not considered in the remaining analysis.

Representative examples of each of the first five supercliques have been presented in the previous three sections. The Type I Cliques 7 and 49 are examples of Supercliques 3 and 4, respectively. Superclique 6 contains Type I cliques which target port TCP/135, similar to Superclique 3, with the difference being that the dominant IAT for cliques from Superclique 6 are in bins 31 and 32, rather than 46, 48, and 49 (for Superclique 3). Cliques 92 and 69 (Type II) are examples of cliques from Superclique 5. The Type III Clique 31 is an example of a clique that belongs to Superclique 1; while Type III Clique 66 is an example of a clique from Superclique 2.

## DISCUSSION AND CONCLUSIONS

We have generated automatically a number of cliques that represent a variety of interesting activities which target the *Leurre.com* environments. We have shown that more than half of the cliques can be easily characterized as one of the three major types identified above. Indeed, in accordance with the supercliques that we manually identified, there are six major classes of activity that the cliquing algorithm has extracted for the time period that we examined (the supercliques). The strong similarities within the supercliques highlight the usefulness of the cliquing algorithm for identifying very particular kinds of traffic observed by the honeypots. Further fine-tuning of the cliquing algorithm may allow these (super)cliques to be automatically generated.

The automatic identification of cliques of the different types outlined in this paper represents a significant contribution to both addressing weaknesses in the original clustering algorithm, as well as highlighting, in a systematic way, the existence of tools with a specific IAT profile. While our analysis has focused only on the *cleanest* cliques (i.e., the ones that represent consistent behaviour in terms of the characteristics we investigated,

1 For example, see http://www.secureworks.com/research/threats/popup-spam/.

such as port sequence, number of virtual hosts targeted, and the targeted platforms), there are many other cliques that contain potentially interesting activities that should be further investigated in the future.

Due to the low interaction nature of the honeypots used by the *Leurre.com* project the majority of the activities observed will relate to different types of scanning (or backscatter), such as that from automatically propagating malware, or scanners which may be cataloguing the existence of various servers around the world, for example. While it is difficult to reach accurate conclusions about the exact nature of the tools which generate the packets collected, we have shown that the cliquing approach adds useful detail to the existing clusters by automatically extracting significant classes of activity from an extremely large dataset.

## REFERENCES


Bomze, I.M., Budinich, M., Pardalos, P. M., and Pelillo, M. (1999) The maximum clique problem, *Handbook of Combinatorial Optimization*, vol. 4. Kluwer Academic Publishers, Boston, MA.

Bomze, I.M., Pelillo, M., and Stix, V. (2000) Approximating the Maximum Weight Clique Using Replicator Dynamics, *IEEE-NN Journal*, vol. 11, no. 6.

Bron, C. and Kerbosch, J. (1973) Algorithm 457: finding all cliques of an undirected graph, *Comm. ACM Press*, vol. 16, no. 9, pp. 575-577, New-York, USA.

Kumlander, D. (2004a) A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search, in proceedings of *Fourth International Conference on Engineering Computational Technology*, Civil-Comp Press, p. 137-138 (an extended abstract - the full paper is published on a CD-ROM).

Kumlander, D. (2004b) An exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring, *Modelling, Computation and Optimization in Information Systems and Management Sciences* (edited by Le Thi Hoai An & Pham Dinh Tao), Hermes Science Publishing, pp. 202-208.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003) A symbolic representation of time series, with implications for streaming algorithms, in proceedings of *Eighth ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, California, USA.

Pavan, M. and Pelillo, M. (2003) A new graph-theoretic approach to clustering and segmentation, in proceedings of *IEEE Conference on Computer Vision and Pattern Recognition*.

Pouget, F. Dacier, M., and Pham, V.H. (2004) Towards a better Understanding of Internet Threats to Enhance Survivability, in proceedings of *International Infrastructure Survivability Workshop (IISW'04)*, Lisbonne, Portugal.

Pouget, F. (2005) Distributed system of Honeypot Sensors: Discrimination and Correlative Analysis of Attack Processes, PhD Thesis from the Ecole Nationale Supérieure des Télécommunications, available through the Eurécom Institute library (www.eurecom.fr).

Pouget, F., Dacier, M., Zimmermann, J., Clark, A., and Mohay, G. (2006) Internet Attack Knowledge Discovery via Clusters and Cliques of Attack Traces, *Journal of Information Assurance and Security*, vol. 1, pp. 21-32.

Spitzner, L. (2003) The Honeynet Project: Trapping the Hackers, *IEEE Security and Privacy*, 1, p. 15.

Zimmermann, J., Clark, A., Mohay, G., Pouget, F., and Dacier, M. (2005) The Use of Packet Inter-Arrival Times for Investigating Unsolicited Internet Traffic, in proceedings of *Systematic Approaches to Digital Forensic Engineering (SADFE)*, Taipei, Taiwan.