Journal of Information Assurance and Security 1 (2006) 21-32

Internet Attack Knowledge Discovery via Clusters and Cliques of Attack Traces

F. Pouget[1], M. Dacier[1], J. Zimmerman[2], A. Clark[2], and G. Mohay[2]

[1]Institut Eurécom BP 193
F-06904 Sophia Antipolis cedex, France
{*pouget, dacier*}*@eurecom.fr*

[2]Queensland University of Technology
126, Margaret Street, Brisbane 4000, QLD Australia
{j.zimmermann, a.clark, g.mohay}@qut.edu.au

***Abstract*:** There is an increasing awareness of the growing influence of organized entities involved in today's Internet attacks. However, there is no easy way to discriminate between the observed malicious activities of script kiddies and professional organizations, for example. For more than two years, the Leurré.com project has collected data on a worldwide scale amenable to such analysis. Previous publications have highlighted the usefulness of so called attack clusters to provide some insight into the different tools used to attack Internet sites. In this paper, we introduce a new notion, namely cliques of clusters, as an automated knowledge discovery method. Cliques provide analysts with some refined information about how, and potentially by whom, attack tools are used. We provide some examples of the kind of information that they can provide. We also address the limitations of the approach by showing that some interesting attack characteristics, namely Inter Arrival Times (IATs) of packets in the attack flows, are only partially taken into account by this approach.

***Keywords*:** honeypots, traffic analysis, Internet attacks, malware, computer forensics

Introduction

Unsolicited traffic on the Internet includes malicious traffic caused by a variety of malicious software such as worms [1] and botnets [2]. It also includes benign traffic and this may be due to a number of different circumstances, including for example software misconfiguration. This unsolicited traffic has been collectively referred to as background radiation by Pang *et al* in [3]. There are broadly speaking two approaches that have been used for studying this traffic in order to identify worms, the first approach using honeypots, the second using so-called Internet telescopes and darknets. We refer the interested reader to [4] for a thorough review of the state of the art with respect to these various approaches. While traffic analysis for this purpose may occur at the packet header level, at the payload level or both, it is noted that there are advantages in using traffic behavior models (relying exclusively or mainly upon packet header information) rather than code signatures (relying upon packet payloads). One reason for this is robustness in the face of obfuscation [2], another is efficiency. There has also been recent work on the use of DNS and ARP activity in a network [5,6] to identify anomalous - possibly worm – activity. Gu *et al.* [7] present a summary of research into the measurement of worm propagation, the modeling of worm propagation, and techniques for early detection and warning. That paper also proposes local detection and response as a more effective way of dealing with blocking the spread of worms across the Internet than the global strategies proposed elsewhere.

The work described in this paper builds upon previous work by the authors in the use of low interactivity honeypots for worm detection and identification using data obtained from a set of honeypot platforms (see [8] for details). This distributed honeypot environment currently consists of 40 platforms located in 25 different countries, covering five continents. Previous work by two of the authors [9, 10] focused on a clustering technique to identify the tools behind the observed attacks using a clustering algorithm detailed in [9].

The present paper focuses on the identification of cluster inter-relationships based on the discovery of cliques. These cluster inter-relationships expose additional common characteristics of the clusters that can be used to gain a better understanding of the modus operandi associated with these tools. To give a concrete example, one would like to know for instance if certain attack tools are used only from some specific regions of the world and target some other specific regions. Similarly, one would like to know if the assumptions made in the context of the study of Darknets or Internet telescopes are valid. Indeed, the underlying assumption of this line of research is that it is possible to know which attacks occur on the Internet as a whole, by extrapolating knowledge obtained by monitoring a sufficiently large block of IP addresses. In the remainder of this paper, we show that, at least in several cases, this hypothesis does not hold.

The main contributions of this paper are:

1. the introduction of the notion of cliques;
2. the use of this notion in several concrete examples;
3. a case study demonstrating the usefulness of the approach and showing that the assumption made by Darknets and Internet telescopes does not always hold; and
4. a discussion of the limitations of the approach when applied to the study of Inter Arrival Times of packets within attack flows.

The rest of this paper is organized as follows. Section 2 briefly presents the Leurré.com project and describes previous work of two of the authors with respect to the

Received November 27, 2005.

Leurré.com honeypot network, and the use of clusters to identify attack tools. Section 3 introduces the notion of cliques of clusters and the distance measures employed in discovering cliques. Section 4 shows how this notion can be applied in practice by means of several examples using our dataset. Section 5 reviews our previous work carried out on the use of IATs to characterize malware and then presents our current results and the problems encountered when trying to use this metric to build cliques. Section 6 presents our conclusions and plans for future work.

Leurré.com, clusters and their limitations

Leurré.com

The Leurré.com network of honeypot sensors currently consists of 40 platforms located in 25 different countries, covering five continents. Most platforms have been active for more than 12 months; the oldest one being active since January 2003. Each platform emulates three virtual machines, each running a different operating system: Windows 98, Windows NT Server and Linux RedHat 7.3. These operating systems are emulated as services using the low-interaction honeypot software *honeyd* [11]. The advantage of this approach is that by design, such honeypots cannot be compromised. Packet level (i.e. tcpdump) traces of the traffic observed at each platform are transferred daily to a central database server. Raw packet level information is enriched, while loaded in the database, with:

1. IP geographical information obtained with NetGeo, MaxMind or IP2location;
2. passive operating system fingerprinting obtained with Disco, p0f and ettercap;
3. TCP level statistics obtained with tcpstat; and
4. DNS reverse lookups results, whois queries.

More details on the database architecture can be found in [12]. Overall, we have observed, up to now, more than 1,240,000 distinct IP addresses targeting our platforms. Those addresses originate from more than 100 different countries. A small number of those addresses have been observed twice, i.e., on two, or more, different days. Most connections consist of a handful of packets, due to the fact that we use low interactions honeypots, where services are only partly emulated. Attack tools are not identified on the basis of a single connection attempt but, instead, by looking at all connections attempted to a number of different hosts. We use the term *Source* to denote an IP address that has been observed on one or several platforms, and for which the inter-arrival time between two consecutive received packets does not exceed a predefined threshold (25 hours).

Using this database, we have shown the validity of this approach for forensic analysis and we have discovered original and interesting observations, related in [10,12]. For the sake of completeness, we briefly recall the clustering technique further below. It is based on the realistic assumption that the attacking tools, if they consist of purely automated deterministic activities, should generate the very same activity fingerprints (a set of parameters discussed below) on all targeted sensors.

Network Influences

Some network disturbances might impact the analysis of malicious activities. They must be carefully taken into account when analyzing attack traces. At the time of writing, we have focused on some particular network effects, namely packet losses, network delays, retransmission, duplicates and forward reordering.

We have developed a technique relying on a particular IP field called the identification field, or IPID, that is normally used in fragment reassembly (see RFC 791 for more details). This field is implemented in various ways, most frequently as a simple incremental counter. The developed algorithm takes into account this property and combines it with the observation of TCP sequence numbers and capture time. In a similar way that Jaiswal et al. used it in [13], we have defined an out-of-sequence (OOS) packet to be a packet whose TCP sequence number (SN) is smaller than the previously observed sequence numbers in that connection. Thanks to this definition, duplicates as well as reordering and retransmission can be detected and removed from the trace under study, as presented on Figure 1.

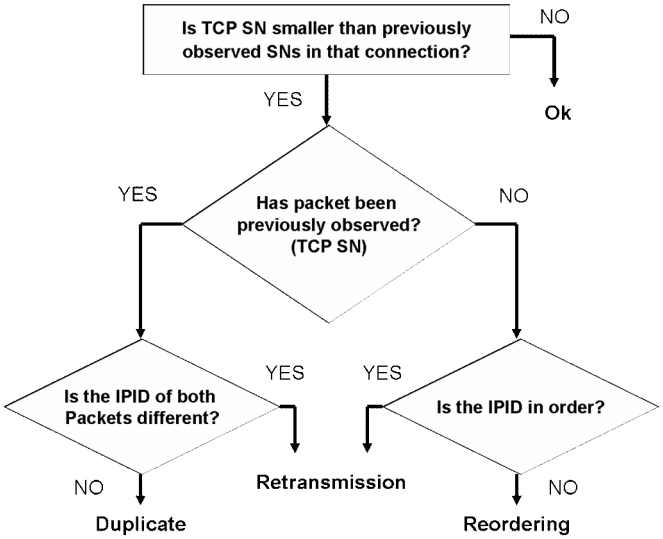

Figure 1. Classification Process of Out-Of-Sequence Packets

Correcting the database is quite straightforward for duplicates and retransmitted packets thanks to this IPID-based method. Unfortunately, packet losses or network delays are biases that cannot be easily fixed[1]. Henceforth we avoid the problem by generalizing some attributes, like the number of packets or the duration of the attacks with regards to these potential network influences. This generalization approach remains realistic and feasible in the scope of our study but its detailed presentation lies outside the scope of this paper. We refer the interested reader to [4] for a comprehensive treatment of this topic.

[1] It is important to note that packet retransmission and loss are different issues and might not be correlated. Some attack tools might implement particular transport layers by themselves, in which case a loss would not be detected and would not imply a retransmission. An interesting summary of the ambiguities in the true semantics of observed traffic has been presented by Paxson in a recent talk [14].

Traffic Clustering

As previously mentioned, we want to define the fingerprint of an attack in terms of a few parameters, in order to easily compare the activities observed on each sensor. We base this step on our own experience of traffic monitoring and on techniques commonly used for network monitoring. We identify the following list of parameters that uniquely define a cluster, or activity fingerprint:

1. the number of targeted virtual machines on the honeypot platform;
2. the sequences of targeted ports. The exact sequence of distinct targeted ports is extracted from the (re)ordered packets sent to one virtual machine;
3. the total number of packets sent by the attacking source to the platform;
4. the number of packets sent by the attacking source to each honeypot virtual machine on the platform;
5. the duration of the activity;
6. the ordering of the activities (in the case where several machines have been targeted, have they been attacked in sequence or in parallel?); and
7. the packet contents (if any) sent by the attacking source (TCP payloads).

We first distinguish parameters with discrete values (parameters 1,2,6) and others where a generalization process is required, as they are more impacted by network losses and delays (parameters 3,4,5). We refer the interested reader to [9] for more details on the clustering technique.

A second step refines the initial grouping by looking at the packet payloads (parameter 7). A Levenshtein-based payload distance is used. If distances are not uniform within a cluster, we split this cluster into smaller and more homogeneous clusters. The output of our clustering engine is a set of clusters where a cluster is defined as follows:

Definition: A *Cluster* groups Sources which had a similar activity fingerprint (similar parameter values) on a Leurré.com platform.

Clustering Limitations

Most of the clusters generated using the technique above are found to be consistent and have helped to understand (to confirm and/or to discover) interesting phenomena of malware activities. Some results have been published in peer-reviewed international conferences [9,10,12].

However, experience has highlighted the following two weaknesses of the approach:

1. It frequently appears that some sets of clusters share other original features. As an illustration, we can cite groups of clusters sharing Time-to-Live (TTL) oddities[2]. In other words, we observe the existence of similarities between clusters that cannot easily be rationalized. The difficulty lies in the fact that these similarities are discriminatory for a small portion of clusters but not for all and, therefore, cannot be used within the previously mentioned clustering algorithm itself.
2. Some clusters are found gathering no more than three sources. In other words, a very small number of IP addresses are found sharing the same activity signature (according to the above mentioned attributes). This might be explained by a too restrictive clustering algorithm, or by the fact that these activities are definitely unusual or very specific. We need to distinguish these two reasons. On one hand, the attributes defined in the clustering algorithm might not be adequate and some clustering criteria should thus be relaxed. On the other hand, very rare activities are also worth being analyzed and understood, as they are of interest for both the administrators of the targeted network and the security community in general.

We intend in the next Sections to move one step further into the resolution of these two issues. Section 3 proposes a technique to address the first issue by identifying such similarities in an automated way through a graph-based approach. Sections 4 and 5 show practical use of this new approach, its benefits as well as its limitations.

Cluster Correlation: Cliques

Introduction

It is important to note that several types of analysis could be applied to the clusters. We distinguish two classes:

1. *Intra-Cluster Analysis:* Within a cluster, the analysis aims at extracting features that are more specific to this cluster than to others, in order to enrich the knowledge and understanding of the phenomenon which has created those traces (root cause of the activity fingerprint).
2. *Inter-Cluster Analysis:* The analysis aims at finding relationships between clusters, and to group those that share common characteristics.

The first type of analysis aims at finding specific features of some attacks. When they are clearly identified, they can be used to improve and check the consistency of the cluster and to improve the matching of new incoming traffic. The second type of analysis aims at checking if the previous features, as well as other properties, are shared by several clusters. We focus in the following on the second type of analysis.

One solution to deal with information extraction from data sets comes from graph and matrix theory. The approach we propose is based on a particular subclass of graphs called *cliques* (also called *complete graphs*, see [15]), and algorithms which aim at extracting dominant sets (maximal weighted cliques).

This technique is applicable to matrices expressing similarities between clusters in a numerical fashion. A similarity matrix is a NxN matrix where each (i,j) element expresses a measure of similarity between the clusters i and j (currently, N≈1000 in our database). Thus, as a preliminary phase, one needs to build a matrix that represents the similarities between any two clusters according to a given criterion. This can be done using the following steps:

1. identify a characteristic of the clusters to be used in the inter-cluster relationship analysis;

[2] For a single Source, we note unusual hops in the TTL values, associated to distinct IPID-counter sequences.

2. define a formal representation for this characteristic (as a vector, for instance);
3. quantify this representation (values to be included in the vector);
4. define a distance metric to evaluate the similarity of the characteristics of two clusters (for instance, vector dot product); and
5. for each pair of clusters, calculate their distance according to this metric and insert the resulting value into the similarity matrix.

Let us assume that we have created such similarity matrices, which model edge-weighted undirected graphs, where each node corresponds to a cluster and the weights of the edges represent the (inverse of the) distances between two nodes (i.e., clusters). We can formalize the problem of discovering inter-cluster relationships as the problem of searching for edge-weighted maximal cliques in the graph of N nodes.

The process is the following: we find a maximal clique in the graph and remove the edges of that clique from the graph; we repeat the process sequentially with the remaining set of vertices and edges, until there remains no non-trivial[3] maximal clique in the graph.

Building the Matrices

Finding maximal cliques in an edge-weighted undirected graph is a classical graph theoretic problem. Because analytic searching for maximal cliques is computationally hard, numerous approximations to the solution have been proposed [16]. For our purposes, we adopt the approximate approach of iteratively finding *dominant sets* of maximally similar nodes in the graph (equivalent to finding maximal cliques) as proposed in [16]. Besides providing an efficient approximation to finding maximal cliques, the framework of dominant sets naturally provides a principled measure of the cohesiveness of a subclass, as well as a measure of node participation in its membership subclass. We now present an overview of dominant sets showing how they can be used for our purpose:

Let the data to be analyzed be represented by an undirected, edge-weighted graph with no self-loops $G=(V,E,A)$ where V is the vertex set, $E \subseteq VxV$ is the edge set and A is the NxN symmetric similarity matrix defined as:

$$a_{ij} = \begin{cases} sim(i,j) & \forall (i,j) \in E \\ 0 & otherwise \end{cases}$$

The *sim* function is computed using any notion of similarity which can be relevant to compare clusters. Some examples are discussed in the next Section.

To discover cliques and to quantize the cohesiveness of a cluster in a clique, we define the "average weighted degree" ($awdeg_S$) of a node in a given subset of vertices S. Let $S \subseteq V$ be a non-empty subset of vertices and $i \in S$ a vertex:

$$awdeg_S(i) = \frac{1}{|S|}\sum_{j \in S} a_{ij}$$

Moreover, for vertices which are not members of the subset, i.e. $j \notin S$, we define Φ_S as:

$$\phi_S(i,j) = a_{ij} - awdeg_S(i)$$

Intuitively, $\Phi_S(i,j)$ measures the similarity between clusters j and i, with respect to the average similarity between i and its neighbors in S. Note that Φ_S can either be positive or negative and that $\Phi_{\{i\}}(i,j) = a_{ij}$ for all j and i belonging to V, with i different from j. Finally, to discover dominant sets in the graph, node weights are assigned to individual vertices[4]. This is done recursively as follows:

Let $S \subseteq V$ be a non-empty subset of vertices and $i \in S$. We denote by S' the set S without the vertex i (ie, S' = S - {i}). The weight of i w.r.t. S is defined as:

$$w_S(i) = \begin{cases} 1 & if |S| = 1 \\ \sum_{j \in S'} \phi_{S'}(j,i) w_{S'}(j) & otherwise \end{cases}$$

The node weight $w_S(i)$ gives a measure of the overall similarity between the cluster corresponding to *i* and the other clusters of S. The total weight of S is then defined as:

$$w(S) = \sum_{i \in S} w_S(i)$$

We are now ready to define the notion of dominant sets:

Definition: A non-empty subset $S \subseteq V$ such that $W(T)>0$ for any non-empty $T \subseteq S$, is said to be *dominant* if:

- $w_S(i)>0$, for $i \in S$ (Internal homogeneity)
- $w_{S \cup \{i\}}(i)<0$, for $i \notin S$ (External inhomogeneity)

Because solving this problem might be infeasible in the general case, we use a continuous optimization technique proposed in [16] which applies replicator dynamics. In other words, solving the problem of extracting dominant sets can be translated into the one of making a particular temporal expression converge (for details, please refer to [16]).

Our algorithm which aims at extracting the maximal set of clusters that share a strong similarity (with respect to the previously built matrix) is then straightforward, and can be summarized by the following pseudo-code:

```
From the weighted graph G=(V,E,A) with N nodes
      WHILE stopping_criterion(G) == FALSE DO
            [1] Extract the dominant set
                from G based on the
                technique referred to above;
            [2] Remove all edges associated
                with the dominant set in G.
      END WHILE
```

The *stopping_criterion()* stops the process when the remaining edges have too small similarity weights or when the dominant set does not contain more than two nodes. It

[3] A non trivial clique is a clique which contains at least three nodes.

[4] Note that the term *weight* is being used to describe both the edge-weights and the node-weights. However, they are two different quantities.

thus prevents us from extracting meaningless dominant sets. Once the dominant sets are found, it suffices to compute their intersections with other dominant sets obtained using other characteristics to determine the new groups of clusters that share strong similarities w.r.t. all analyses.

Experimental Validation of the notion of Cliques

Definition of eight sample matrices

This technique has been successfully applied with eight different analysis matrices listed hereafter:

- *A_Geo*: Distribution of attacking countries; this distance matrix presents a comparison of countries where the attacking IPs are located. Cliques obtained using this matrix highlight clusters presenting common distribution of attacking countries. For example, it leads to the following observation: certain tools have been seen launched most frequently from a very limited number of countries only.
- *A_Env*: Distribution of targeted environments; this matrix presents a comparison of the environments targeted by the attacks. Cliques obtained using this matrix reveal that some platforms are attacked by some tools that are not observed elsewhere. Their existence demonstrates a limitation of the approaches which rely on Internet telescopes and darknets for extrapolation.
- *A_OSs*: Distribution of attacking operating systems among clusters. It answers the following question: which attack activities are launched from the same sets of operating systems?
- *A_IPprox*: Attacking Source IP Proximities. We compute here the distance between an attacking IP address and its target IP address. It leads to groups of clusters that, quite likely, share similar propagation strategies such as, for instance, a bias in favour of the propagation within the same class C, etc.
- *A_TLDs*: Distribution of Top-Level Domains (DNS) among clusters.
- *A_Hostnames*: Distribution of attacking machine types among clusters (servers, routers, dsl home machines, etc).
- *A_CommonIPs*: Shared common IPv4 addresses between clusters.
- *A_SAX*: Temporal evolution over weeks (Time Series analysis) to compare the temporal trends of each attack process characterized by a cluster over a period of several days. The method is based on the notion of so called 'time signature' of the clusters, introduced in [17] and obtained using a recent time series analysis method called the Symbolic Aggregate approximation (SAX).

In the following, for the sake of illustration, we describe in more detail the building of the matrices *A_Env*, *A_IPprox* and *A_CommonIPs* as well as the results obtained with our dataset. We then show in Section 4.5 how these individual results can be combined to deduce some interesting information from these simple traces.

Results with the "A_Env" Matrix

The *A_Env* matrix aims at finding correlation between clusters that have mostly been observed on a limited number of environments as opposed to the ones that have been observed on all platforms in a homogeneous way. Therefore, we compute for each cluster, for each environment, the ratio (in percentage) between the amount of traces due to that cluster on that environment over the total amount of traces for the same cluster observed on all platforms. The results are stored in C vectors of size P, where C is the total number of clusters and P is the number of platforms. The sum of all P values in each vector equals 100 by definition. For each of these C vectors, we apply a peak picking algorithm, which aims at selecting the most frequent platforms for each cluster.

For each vector, all peaks that are μ times more intense than the average distribution are extracted and ordered in decreasing order[5]. All lists of peaks for all vectors are then compared by pairs. A distance of 1 between two clusters indicates a complete match of their ordered list of peaks, otherwise the distance is set to zero.

With *A_Env*, the *dominant set extraction algorithm* generates 12 cliques. They are shown in Table 1 with the *Number of Clusters*, the *Clique Coverage* and the *Peaks*. The Clique Coverage value provides the ratio between the number of attack traces[6] found in all clusters included in a given clique and the total number of attack traces found in the whole dataset. It gives a good indicator of the relative importance of the clique in terms of volume of attack traces. In the Peaks column, we provide the list of platform identifiers that are mostly attacked by the clusters found in the clique.

As an illustration, Table 1 shows that 30 distinct fingerprints (or clusters) are specific to platform 20, while 28 are only observed on platform 6, etc. This confirms the distinctive nature of some attacks we had noticed manually and contradicts the assumption that traffic observed by an Internet Telescope is representative of all the Internet traffic. It is important to note, though, that a large number of attacks are still observed on most platforms. For these ones, Internet telescopes provide a suitable environment to study for example their propagation strategies.

Clique Id	# Clusters	Clique Coverage	Peaks
ID 1	30	4.62	{20}
ID 2	28	2.39	{6}
ID 3	20	3.00	{20,8}
ID 4	18	2.39	{32}
ID 5	14	2.01	{20,25}
ID 6	26	3.88	{25}
ID 7	43	6.42	{6,31}
ID 8	10	0.97	{8,6}
ID 9	8	0.93	{6,8}

[5] We consider μ = 2 in this case, as most of the distribution are not uniformly distributed and they present clear peak activities.

[6] To be exact, we should use the term *Large session*, as defined in the Leurré.com jargon, instead of attack traces.

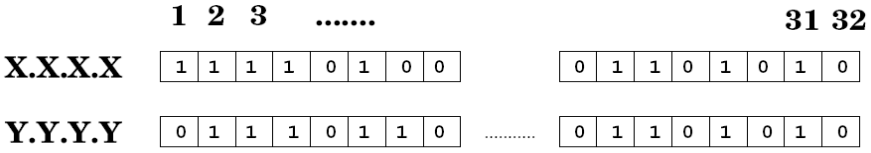

Figure 2. Distance between IP addresses

Clique Id	# Clusters	Clique Coverage	Peaks
ID 10	14	1.6	{23}
ID 11	12	2.28	{10}
ID 12	5	0.42	{25,20,36 }

Table 1. Cliques obtained using Matrix *A_Env*

Results with the "A_IPprox" Matrix

Some malware is known to favour the propagation over local networks, changing, for instance, some of the last IP bits of the compromised machines to select a new target. Code Red II, for instance, implements such a strategy [18]. In 1/8th of the time, this worm generates a random IP address not within any range of the already compromised IP address. In half of the time, it will stay within the same class A range as the compromised IP address and in 3/8th of the time, it will stay within the same class B range as the attacking IP address. Therefore, this worm has a particular signature in terms of IP distances. Clusters corresponding to Code Red activities should have this kind of ratio in common when looking at the distance between the attacker and attacked. We know that such behaviour should be observed for Code Red II because it has been carefully analysed (and modeled) by security researchers [18]. For the same reasons, we also know that the Zotob worm propagates by keeping the first 2 bytes and tries to connect to random IP addresses within the same B-class (255.255.0.0) as the compromised machine. However, reverse engineering worm code is a time-consuming task. We would like to find a way to identify systematically such bias in the propagation without having to rely on code analysis of the malware. To provide such answers, the matrix *A_IPprox* is built to determine if some clusters share strong similarities with respect to the distance between the attacking source IP address and the targeted virtual machine IP address.

In order to compute the distance between IP addresses, we use a specific function that returns the first bit position from which two IPv4 addresses IP1 and IP2 differ, with a Big-Endian approach. An illustration is presented in Figure 2. The first bit which differs between IP1= X.X.X.X and IP2=Y.Y.Y.Y is at position 1, so the distance is 1. This operation is performed for each pair of IPs (Attacking Source / Targeted Virtual Machine) within a Cluster and the considered vector is simply the distribution of these values over all traces in the cluster.

Using the *A_IPprox* matrix as input, the dominant set extraction algorithm generates 53 cliques. Of these, 21 contain at least 30 clusters. They correspond to several combinations of peaks for distances of values 1, 5, 9, 17, 25 and 31. This interesting result indicates that many activities observed from the distributed honeypot architecture present quite basic propagation features limited to the /8, /16, /24 or even /30 subnets of the victims. We are currently investigating other definitions of IP distances to provide a refined analysis of these shared propagation strategies.

Results with the "A_CommonIPs" Matrix

Another interesting similarity matrix is the one that reveals the percentage of common IP addresses between clusters. It can be imagined that an address X.X.X.X first launches an attack A1 (for example a scan), and then comes back a few days later to run attack A2. The very same address will appear in the clusters corresponding to attack A1 and attack A2. Identifying the fact that these two clusters have a large fraction of IPs in common will help us identifying the *modus operandi* of attack A2, which will appear to be always preceded by attack A1.

A practical case of such a situation exists with the worms that take advantage of backdoors left open by other worms. A famous example is the *Dabber* worm that exploits the same vulnerability as the *Sasser* worm in order to spread. It uses a backdoor installed by the *Sasser-FTP* exploit application to burrow into a PC. Then, it removes *Sasser*, and installs a server on the infected machine to keep propagating. According to this scenario, we can expect that clusters associated with *Sasser* and *Dabber* may share common IPs. They will first be seen propagating the *Sasser* worm and, then, propagating *Dabber*.

We have built the similarity matrix *A_CommonIPs* to provide a systematic way to answer that kind of question. Let A and B be two distinct sets of IP addresses. The distance is defined as:

$$sim_{IP}(A,B) = \frac{|A \cap B|}{|A \cup B| - |A \cap B|}$$

As we can see, this is a different type of distance, based on the cardinality of sets as opposed to the other ones that were considering the distributions of some parameters within the clusters. The extraction of dominant sets reveals that the scenario mentioned at the beginning of this section has not been observed so far. It may be that no worm exhibiting such behaviour is virulent enough to appear in the limited number of addresses that we have at our disposal, or that the influence of temporary IP addresses is so prevalent that it hides the fact that the same machines are coming again. We are currently considering how to generalize this analysis by looking at common blocks of addresses instead of common IP addresses. However, it is worth noting that we still obtain around 50 cliques formed by numerous clusters which are of very small sizes (less than 5 Sources per cluster). They correspond to very particular activity phenomena that are, in almost all cases, the artefact of misconfigured management

servers. As an illustration, it has been found that a single HP Openview Server (or, more precisely, a *HP Systems Insight Manager HPSIM*) has periodically been scanning machines in a given network, using different layer 3 protocols and transport ports (UDP 161: SNMP, TCP 280: http management, etc). This server has been observed for five months during discontinuous periods of varying lengths, from a few hours to a few days. Since the length of a session as well as the number of packets are key elements of the definition of a cluster, this IP has been put in various clusters as a function of the length of its continuous activity. Later on, these various clusters have been brought back together by means of this similarity matrix.

An illustrative Case Study

The three previous sections have described three matrices which have been built to perform some analyses on the clusters. We want to show in this Section, by means of a short example, that the intersection of several analyses can also lead to interesting findings. Therefore, we focus on two dominant sets that have been defined in Table 1 with the respective identifiers 8 and 9. These cliques group together all clusters, 10 for clique 8 and 8 for clique 9, that have targeted both platforms 6 and 8. The first platform is located in an European industry network, while the second one runs in an academic network in Asia. Investigating the intersections of these cliques with the ones obtained by means of the other analyses leads to the following results:

1. all 18 clusters are also found within a single clique obtained by means of the *A_IPprox* matrix. That clique corresponds to the single peak {24}. In other words, it means that all attacking sources are located in the same /24 network as their targets. However, the targets are located in two different /24 networks. Thus, all sources belong either to the same /24 as sensor 6 or in the same /24 network as cluster 8. This is a very striking result coming from the simple intersection of the cliques obtained with these two matrices;
2. despite the fact that the attackers come from two limited sets of IP addresses, all IPs found in these 18 clusters are different. Indeed, none of these clusters show up in the matrix *A_CommonIPs*; and
3. none of these 18 clusters belongs to the cliques obtained using the *A_SAX* matrix. This means that the involved clusters do not share a similar temporal evolution. It is therefore quite likely that they represent different kinds of activity and are not the various symptoms of a single worm present in a given limited period of time.

By digging into the definitions of the clusters involved, one can easily find that all attacks found in these clusters have targeted the two virtual machines emulating Windows. None of the attacks have targeted the Linux emulated platform. They all have targeted port 135 (Microsoft Remote Procedure Call), but in many different ways (in terms of duration, payload, etc). Interestingly, these attacks have not been observed on any other platform than these two. These traces cannot be interpreted as non malicious *radiation noise*, as there is no service running on port 135 of the virtual machines and they do not respond to multicast requests. There is no regular temporal pattern between the different attacks. It is thus not a process trying to probe the Windows machines in a periodic manner. It could however be a monitoring process distributed over different machines with random time intervals.

This example leads to the following conjecture. It is quite likely that a hostile *botnet*[7] is running on each of the class C networks where sensors 6 and 8 are installed. These two botnets are similar in the sense that they do not launch attacks against the whole world but against their own /24 only. These botnets at several occasions have run different types of attacks but have always focused on port 135. Surprisingly enough, these attacks have not been observed elsewhere afterwards. It is thus possible that these botnets are used as testbeds for malicious users in the process of developing a new worm against port 135. What we see on these networks would be the preliminary versions of the worm, not yet ready to be deployed on a large scale. It would be interesting to pursue this investigation by looking for a new type of cluster that would have been seen initially on one of these platforms and then in the rest of the world afterwards. This would provide one more argument in favour of the usage of these networks as small scale test networks used before launching mature and stable new worms in the wild.

In the next Section, we investigate another interesting characteristic that has proved to be useful to characterize attack traces, namely Inter Arrival Time of packets in attack flows. We show how this notion can also be used within the very same framework.

IATS and cliques

Introduction

In previous work [20], we have investigated the use of packet inter-arrival times (IATs) to analyse the Leurré.com dataset. In that work we were interested in the IATs between packets sent from one attacker to one virtual machine on a given platform (i.e. a so called "*tiny session*" in the Leurré.com jargon). Clearly, IATs are influenced by the network, mostly by the varying latency but also by random perturbations. However, our intuition was that they could also help us identifying some pattern of activities that would otherwise remain hidden. In [20] we show, for instance, some unexpected IAT values present in a large number of tiny sessions. In one case, machines are regularly sending packets every 8 hours while in another case we observe a very precise IAT value of 9754 seconds (i.e. 2 hours, 42 minutes and 34 seconds) between two packets in many different sessions. We refer the interested reader to [20] for a detailed treatment of these observations. While there was no evidence to suggest that these IAT peaks were indicative of some malicious behaviour (instead appearing to be caused by misconfigured devices), the approach provided a useful technique for classifying large volumes of traffic most likely

[7] From [19]: "[…]A *botnet* is a structure consisting of many compromised machines which can be remotely managed (in general from an Internet Relay Chat IRC channel) […]".

caused by the same software. It is worth stressing the fact that these results had been obtained without taking the notion of clusters into account at all. At this stage, we focus on the question whether existing IATs peaks could be used to group several different clusters together. If yes, it would mean that the propagation strategy of these tools was visible through the IAT peaks and common to several of them. This would be a very interesting input for those carrying out forensics analysis of these phenomena. In the following subsection, we show how we have created cliques of clusters based on IATs values and the results we have obtained.

IAT vectors, distance and matrices

In this analysis, we only consider IATs comprised between 5 minutes (= 300 seconds) and 25 hours (= 90000 seconds), the maximum IAT value allowed by the session model. The 5 minute threshold has been chosen as a way to eliminate all possible influences of the network perturbations. Indeed, we assume that only IATs longer than 5 minutes can be considered as characteristic of the monitored traffic; shorter IATs may depend on other network artefacts such as congestion, packet losses, transmission latencies etc. These require a different analysis approach which is subject to future work. Of course, large IATs can still be influenced by the network and this is something that we will need to take care of.

IAT Vector

As explained before, the very first step in order to generate cliques of clusters is to define for each of them the characteristic we are interested in, by means of a vector. In this case, the approach taken consists of keeping for each cluster the list of IAT values that are predominant in the traces of that cluster. This is achieved using a simple "peak picking" algorithm. We define IAT peaks as values corresponding to large value variations in the statistical distribution of IATs. On a normed scale, the 1st derivative of a distribution with peaks shall thus consist of near-zero values, with wide oscillations appearing for peak values. Furthermore, we keep the sole peaks that exceed a certain threshold, defined as a multiple of the distribution's standard deviation. Among other things, this enables us to discard all peaks that appear in sessions where only two packets have been observed (ie having a single IAT value).

More specifically, our "peak-picking" algorithm is made of the following steps, for each cluster:

1. initially, the set P of IATs which correspond to peaks is void;
2. calculate D, the statistical distribution of IATs in the tiny_sessions of the given cluster, and its first derivative D'. Figure 3 shows an example of D where the granularity of the IATs is of 5 seconds. This leads to a maximal value of 18000 since the largest possible IAT value is 90000 seconds;
3. calculate C, the convex envelope of D' represented in polar coordinates. C corresponds to the most prominent peaks in D' and thus to those in D. Figure 4 represents C and D' corresponding to the curve D shown on Figure 3;
4. add C to the set of IAT peaks P and set the values of D' corresponding to these IATs to zero;
5. repeat steps 3 and 4 to identify more peaks;
6. filter P to retain only those peaks whose height exceeds a certain multiple of the standard deviation of D. In Figure 5, we represent with the "+" symbol the peaks that are automatically selected for that curve.

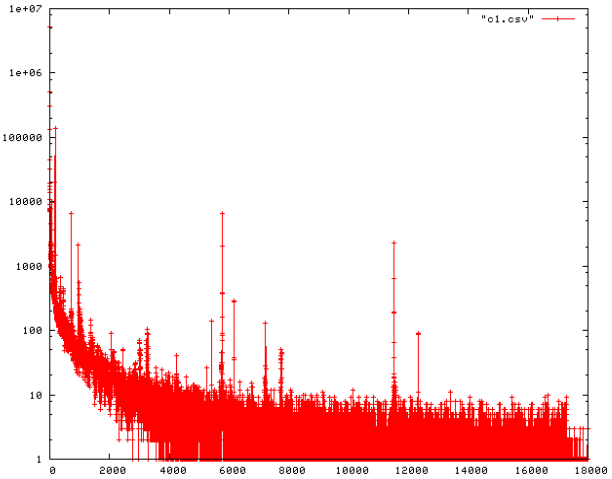

Figure 3. Distribution of IATs (5 seconds interval)

The results of this algorithm can be influenced by several parameters, namely:

- the resolution used to generate the distribution D at step 2, denoted as r and equal to 5 seconds in the previous example;
- the number of rounds, i.e. the number of times steps 3 and 4 are performed, denoted as n;
- the threshold value used at step 6, denoted as t.

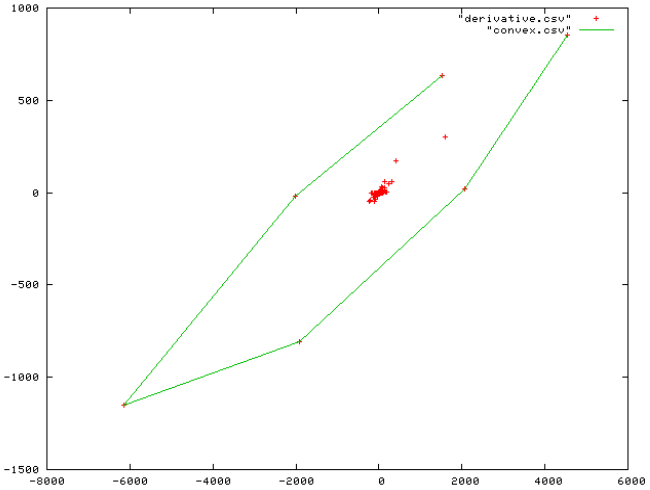

Figure 4. Convex Envelope, in polar coordinates, of the first derivative of the curve given in Figure 3.

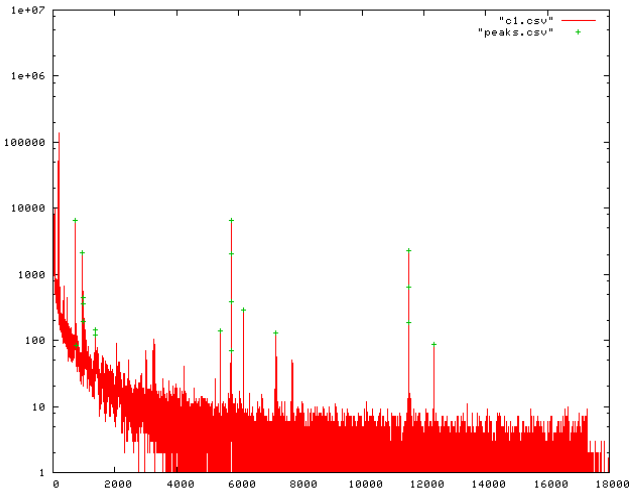

Figure 5. Peaks of *D* are indicated by the "+" symbols

The results discussed here after are based on the following values: r=5, n=4 and t=1.5. The distribution resolution of r=5 was selected based on our previous experience [20], which showed that most peaks tend to appear centered around an IAT value +/- 2 seconds. The values used for n and t were chosen empirically after a series of tests.

IAT Distance

Using the process described above, we obtain a vector of IATs that correspond to peaks for each processed cluster. To evaluate similarity of clusters in terms of IAT peaks, one has thus to define a distance metric function for each pair of vectors. Various functions can be considered, ranging from a plain vector dot product to elaborate shape-matching descriptors. Defining objective criteria for a sensible distance function is a nontrivial problem, especially for vectors of variable lengths.

In the following, we use simply the percentage of the peaks detected in two clusters that are common to the two clusters. If we denote P(c) the list of IAT peaks that appear as peaks in the cluster c, then the distance, in terms of IATs peaks, between two clusters c1 and c2 is thus obtained by the following formula:

$$d(c1,c2) = | P(c1) \cap P(c2) | * 100 / [|P(c1)| + |P(c2)|]$$

High similarity of clusters in terms of IAT peaks thus leads to high values of d. We note that this distance metric disregards the actual number of IAT occurrences, i.e., the heights of the peaks, and considers only the values of IATs which correspond to peaks.

IAT matrix

To accelerate the clique generation, we discard the clusters for which no IAT peak was detected by the peak-picking algorithm. Results show that the values of d obtained are generally low: mostly in the 0-33% range. Strong similarities indicated by high values of d (from 66% up to 100%) appear for pairs of clusters that correspond to very specific traffic such as series of ICMP packets sent at regular intervals. With the parameters used (i.e. *n=4*, *r=5* and *t=1,5*), we obtain 54 clusters that have at least one high similarity with another one. Within these 54 clusters, 35 correspond to ICMP traffic. With two exceptions, the IAT values that correspond to peaks in these clusters are various multiples of 5 minutes, comprised between 300 seconds (the minimum IAT considered) and 7200 seconds. Three clusters also include multiples of 5 minutes +/- 5 seconds, i.e. clusters which include wider peaks that spread across two columns in our 5 second-based IAT distribution. Finally, only three of these 35 ICMP clusters are remarkable: one ICMP cluster contains also TCP traffic directed to the TCP port 4280 (in addition to ICMP traffic) and two pure ICMP clusters feature IAT peaks that are not multiples of 5 minutes +/- 5 seconds (namely, 355, 410, 420, 445, and 390 seconds).

The application of the cliques highlighted a limitation of the distance vector as it had been chosen. Indeed, cliques have grouped together clusters that were sharing the same peaks, for instance the last two, but, of course, they failed in showing that most peaks were multiples of 5 minutes. More explicitly, if cluster C1 has for example three IATs peaks at 20, 25 and 30 minutes while C2 has two peaks at 10 and 15 minutes, they will never be grouped together but, intuitively, one might wonder if these 5 minutes interval are not worth being reported to the analysts. Investigating the pattern between IAT values is part of our ongoing work.

The observations concerning the remaining non-ICMP clusters can be summarised as follows:

- 5 clusters correspond to traffic directed to TCP ports 135 and/or 445 (5 clusters) and three to TCP port 80. The traffic type of two clusters remain unidentified. All these clusters but one TCP445 cluster have IAT peaks which are multiples of 5 minutes . Here too, for the reasons explained above, they have not been grouped together by the clique system despite their apparent similarities; and
- 8 clusters contain Windows Messenger related traffic (1026UDP), five of which feature numerous peaks (3 clusters feature only one peak each). All IAT peaks in these clusters correspond to various values spread in the 2105 – 3770 seconds range ; 2105 seconds being the only one close to a multiple of 5 minutes. These clusters did not share enough peaks to be grouped together into any given clique. However, the fact that they all target the same types of ports leads us to believe that the existence of these peaks, yet different, might be of interest to the people who analyse these threats.

We can derive three important conclusions from these results. First of all, we have to acknowledge the fact that the cliques did not deliver the results that we were expecting with respect to that specific characteristic. However, and this is the second contribution, the manual inspection of the IAT matrices has highlighted groups of clusters that clearly have some common characteristics in terms of IATs, yet not formalized in an appropriate way to be used by the clique algorithms. We do hope though to find a better way to express in a near future these IATs features in a such way that the clusters would eventually end up in real cliques automatically. Finally, as a follow up to the previous remark,

we may have to revisit the way we do the fusion of the results of the various matrices. As of now, we simply rely on the intersection of the cliques. Previous results show that, sometimes, the mere existence of a high similarity between two clusters, without the existence per se of a clique, is something that is worth considering. This is something that we are currently investigating as well.

Conclusions

In this paper, we have introduced the notion of cliques as an automated way to find interesting information about similarities in the *modus operandi* of several, apparently unrelated, attack tools. We have proposed a generic framework that enables us not only to easily add as many different view points as possible but also to perform a simple, yet efficient, aggregation of the results obtained by means of various types of analyses. The usefulness of the approach has been validated experimentally and results have been detailed in the paper. A simple, yet illustrative, case study has been proposed. Last but not least, we have seen how to reconcile the study on the IATs within that framework. Here too, experimental results have shown the merits of the approach as well as its limits.

The results described herein are extremely promising in the sense that they offer an easy way for analysts to extract as much information as possible from traces obtained on low interaction honeypots which are very easy to deploy on a large scale. This framework, thanks to the enriched information found in the Leurré.com database, provides a semantically rich environment to interpret the data and to better understand the attack threats found on the Internet.

Acknowledgments

The above work was supported in part by a joint French-Australian Science and Technology (FAST) Programme grant and an ARC International Linkage Postdoctoral Fellowship grant.

References

[1] J. Nazario, "*Defense and Detection Strategies against Internet Worms*". Artech House, 2004.

[2] D. Geer, "Malicious Bots Threaten Network Security," *Computer*, vol. 38, pp. 18-20, 2005.

[3] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, L. Peterson, "Characteristics of internet background radiation," in Proceedings of *4th ACM SIGCOMM conference on Internet measurement,* Taormina, Sicily, Italy, 2004.

[4] F. Pouget, *Distributed system of Honeypot Sensors: Discrimination and Correlative Analysis of Attack Processes*, PhD Thesis from the Ecole Nationale Supérieure des Télécommunications, January 23, 2005, available through the Eurécom Institute library (www.eurecom.fr).

[5] D. Whyte, E. Kranakis, P.C. van Oorschot, "DNS-based Detection of Scanning Worms in an Enterprise Network" In Proceedings of *Network and Distributed System Security Symposium (NDSS'05).* San Diego, 2005.

[6] D. Whyte, P.C. van Oorschot, E. Kranakis, "Detecting Intra-enterprise Scanning Worms Based on Address Resolution" In Proceedings of the *ACSAC 2005*.

[7] G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, G. Riley, "Worm Detection, Early Warning and Response Based on Local Victim Information," presented at *20th Annual Computer Security Applications Conference (ACSAC 2004)*, Tucson, Arizona, 2004.

[8] The Leurré.com home page, in http://www.leurrecom.org.

[9] F. Pouget, M. Dacier, "Honeypot-based forensics" In Proceedings of *AusCERT Asia Pacific Information Technology Security Conference 2004 (AusCERT2004)*, 2004.

[10] F. Pouget, M. Dacier, V.H. Pham, "Leurre.com: On the Advantages of Deploying a Large Scale Distributed Honeypot Platform" In Proceedings of *E-Crime and Computer Conference 2005.(ECCE'05)* Monaco, March 2005.

[11] N. Provos,"A Virtual Honeypot Framework" In Proceedings of the *13th USENIX Security Symposium 2004*, Boston USA, July 2004.

[12] F. Pouget, M. Dacier, H. Debar, "Honeynets: Foundations for the Development of Early Warning Systems", In Proceedings of *the Cyberspace Security and Defense: Research Issues*, Publisher Springler-Verlag, LNCS, NATO ARW Series, 2005.

[13] S. Jaiswal, G. Iannacconne, C. Diot., D.F. Towsley, "Inferring TCP Connection Characteristics Through Passive Measurements", In Proceedings of *IEEE Infocom 2004*, Hong-Kong, March 2004.

[14] V. Paxson, "Strategies for sound Internet measurement", In Proceedings of *4th ACM SIGCOMM Conference on Internet Measurement IMC'04*, Italy, 2004.

[15] C. Bron, J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph", *Comm. ACM Press*, Vol. 16, Nb 9, 0001-0782, pp. 575-577, New-York, USA, 1973.

[16] M. Pavan, M. Pelillo, "A new graph-theoretic approach to clustering and segmentation", In Proceedings of *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[17] J. Lin, E. Keogh, S. Lonardi, B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms", In Proceedings of *8th ACM SIGMOD workshop on Research Issues in data mining and knowledge discovery*, California, USA, 2003.

[18] C.C. Zou, W. Gong, D. Towsley, "Code Red Worm Propagation Modeling and Analysis", In Proceedings of *ACM CCS'02*, Washington DC, USA, November 2002.

[19] B. McCarty, "Botnets: Big and Bigger", In *IEEE Security and Privacy*, 1(4):87-90, 2003.

[20] J. Zimmermann, A. Clark, G. Mohay, F. Pouget, M. Dacier, "The Use of Packet Inter-Arrival Times for Investigating Unsolicited Internet Traffic", In Proceedings of *the First International Workshop on*

Systematic Approaches to Digital Forensic Engineering SADFE 2005, Taiwan, November 2005.[1]Nazario J., "*Defense and Detection Strategies against Internet Worms*". Artech House, 2004.

Author Biographies

Fabien Pouget recently completed his Ph.D. at the Institut Eurecom, France. He received his master of Science from the Ecole Nationale Superie*ure des Telecommunications (ENST) in 2002 after* having worked as internship student in the IBM Research laboratory in Zurich, Switzerland. He joined the Network Security Team (nsteam) at Eurecom the same year. His research and teaching interests include computer and network security. He is involved in many projects on intrusion detection systems and honeypots and his PhD subject deals with alert correlation.

Marc Dacier is a professor at the Eurecom Institute. He also is an associate professor at the University of Liege in Belgium. *From 1996 until 2002, he worked at IBM Research as t*he manager of the Global Security Analysis Lab. In 1998, he co-founded with K. Jackson the "Recent Advances on Intrusion Detection" Symposium (RAID). He is now chairing its steering committee. He also was the co-director, with Brian Randell from the University of Newcsastle, of the MAFTIA European Project (Malicious and Accidental Fault Tolerance for Internet Applications). He serves on the program committees of major security and dependability conferences and is a member of the steering committee of the "European Symposium on Research for Computer Security" (ESORICS). He serves on the editorial board of the IEEE Transactions on Dependable and Secure Computing (TDSC). He is leading the Leurre.com project (www.leurrecom.org), a worldwide distributed honeypot system deployed in more than 20 different countries. His research and teaching interests include computer and network security, intrusion detection, network and system management. He is the author of numerous international publications and several patents.

Jacob Zimmermann is an postdoctoral fellow with the Information Security Institute at Queensland University of Technology in Brisbane, Australia. He completed his PhD on policy-based intrusion detection using noninterference in December 2003 at Supelec, France and joined the ISI in January 2005. His current research topics include policy-based intrusion detection, exploitation of vulnerabilities and honeypot traffic analysis.

Andrew Clark is a researcher with the Information Security Institute at Queensland University of Technology in Brisbane, Australia where he has been employed since 1992. He obtained his PhD in 1998 in the field of information security and has been involved with numerous projects in that field with industry partners. He is a member of the program committee of the "Recent Advances in Intrusion Detection" (RAID) conference. He currently supervises a number of students researching in the areas of intrusion detection and computer forensics. His current research interests are in the areas of honeypots, wireless network security and intrusion detection.

George Mohay is an Adjunct Professor in the Information Security Institute at the Queensland University of Technology, Brisbane, Australia. Prior to that he had been Head of the School of Computing Science and Software Engineering from 1992 to 2002. His current research interests lie in the areas of computer security, intrusion detection, and computer forensics. He has worked as a visiting researcher while on sabbatical leave at Stanford University in 1981, Loughborough University in 1986, Bristol University in 1990 and the Australian National University in 2000. He graduated BSc(Hons) (UWA) in 1966 and PhD (Monash) in 1970. He supervises and has supervised PhD and Masters students in the above areas. He is currently involved as chief investigator in a number of related funded research projects: in the area of computer forensics with Australia's DSTO (Defence Science and Technology Organization), a DEST FAST sponsored project on Internet security, and an ARC sponsored project on intrusion detection. Since 1995 he has been involved in over ten funded projects, including a number of other ARC projects. His publications include the recently published book Computer and Intrusion Forensics. He is a program committee member for a number of international conferences: RAID - Recent Advances in Intrusion Detection, ACM CCS (Computer and Communications Security), ACISP (Australasian Conference on Information Security and Privacy), SADFE (Systematic Approaches to Digital Forensic Engineering) and ICA3PP (International Conference on Algorithms and Architectures for Parallel Processing); and he is program committee co-chair for the research stream of the annual AusCERT security conference. He is a member of the ACM, and of the IEEE Computer Society.