A Network Security Analysis Tool

Tuva Stang[†], Fahimeh Pourbayat[†], Mark Burgess[†],
Geoffrey Canright[‡], Kenth Engø[‡] and Åsmund Weltzien[‡]
Oslo University College, Norway[†] and Telenor Research, Oslo, Norway[‡]

Abstract

Archipelago is system analysis and visualization tool which implements several methods of resource and security analysis for human-computer networks; this includes physical networks, social networks, knowledge networks and networks of clues in a forensic analysis. Access control, intrusions and social engineering can be discussed in the framework of graphical and information theoretical relationships. Groups of users and shared objects, such as files or conversations, provide communications channels for the spread of both authorized and un-authorized information. We present a Java based analysis tool that evaluates numerical criteria for the security of such systems and implements algorithms for finding the vulnerable points.

1 Introduction

Network security can be discussed from many viewpoints. Some discussions focus entirely upon the technologies that protect individual system transactions, e.g. authentication methods, ciphers and tunnels. Less attention has been given to the matter of *security management*, where a general theoretical framework has been lacking. In this work, we explore two theoretical methods to estimate *systemic security*, as opposed to system component security. We describe a tool (Archipelago) for scanning systems, calculating and visualizing the data and testing the results.

Our paper starts with the assumption that security is a property of an *entire system*[1] and that covert channels, such as social chatter and personal meetings, are often viable ways to work around so-called strong security mechanisms. File access security is a generic representation of communication flow around a system, and we use it as a way of discussing several other problems. Other issues like social engineering have previously been notoriously difficult to address in quantitative terms, but fit easily into our discussion. We have made some progress in this area by applying graph theoretical techniques to the analysis of systems[2]. In this paper we implement a tool for using these techniques and demonstrate its use in a number of examples.

The paper begins with a brief discussion of the graph-theoretical model of security, and how it is used to represent associations that lead to the possible communication of data. Next we consider how complex graphs can be easily represented in a simplified visual form. The purpose of this is to shed light on the logical structure of the graph, rather than its raw topological structure. We describe a method of eigenvector centrality for ranking nodes according to their importance, and explain how this can be used to organize the graph into a logical structure. Finally, we discuss the problem of how easily information can flow through a system and find criteria for total penetration of information.

2 Graphs

A graph is a set of nodes joined together by edges or arcs. Graph theoretical methods have long been used to discuss issues in computer security[3, 4], typically trust relationships and restricted information flows (privacy). To our knowledge, no one has considered graphical methods as a practical tool for performing a partially automated analysis of real computer system security. Computer systems can form relatively large graphs. The Internet is perhaps the largest graph that has ever been studied, and much research has been directed at analyzing the flow of information through it. Research shows that the Internet[5] and the Web[6] (the latter viewed as a directed graph) each have a power-law degree distribution. Such a distribution is characteristic[7, 8, 9] of a self-organized network, such as a social network, rather than a purely technological one. Increasingly we see technology being deployed in a pattern that mimics social networks, as humans bind together different technologies, such as the Internet, the telephone system and verbal communication.

Social networks have may interesting features, but a special feature is that they do not always have a well defined centre, or point of origin; this makes them highly robust to failure, but also extremely transparent to attack[10]. A question of particular interest to a computer security analyst, or even a system administrator deploying resources is: can we identify likely points of attack in a general network of associations, and use this information to build analytical tools for securing human-computer systems?

3 Associations

Users relate themselves to one another by file sharing, peer groups, friends, message exchange, etc. Every such connection represents a potential information flow. An analysis of these can be useful in several instances:

- For finding the weakest points of a security infrastructure for preventative measures.
- In forensic analysis of breaches, to trace the impact of radiated damage at a particular point, or to trace back to the possible source.

Communication takes place over many channels, some of which are controlled and others that are *covert*. A covert channel is a pathway for information that is not subject to security controls.

Our basic model is of a number of *users*, related by associations that are mediated by human-computer *resources*. The graphs we discuss in this paper normally represent a single organization or computer system. We do not draw any nodes for outsiders; rather we shall view outsiders as a kind of reservoir of potential danger in which our organization is immersed.

In the simplest case, we can imagine that users have access to a number of files. Overlapping access to files allow information to be passed from user to user: this is a channel for information flow. For example, consider a set of F files, shared by U users (fig. 1).

Here we see two kinds of object (a bi-partite graph), connected by links that represent associations. A bipartite form is useful for theoretical discussions, but in a graphical tool it leads to too much ‘mess’ on screen. Bi-partite graphs have been examined before to provide a framework for discussing security[11]. We can eliminate the non-user nodes by simply colouring the links to distinguish them, or keeping their character solely for look-up in a database.

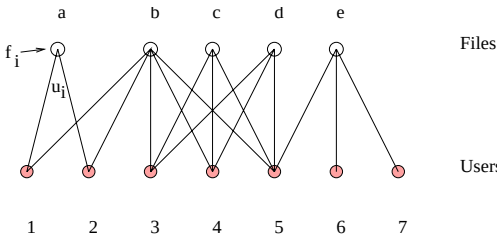

Figure 1: Users (dark spots) are associated with one another through resources (light spots) that they share access to. Each light spot contains f_i files or sub-channels and defines a group i, through its association with u_i links to users. In computer parlance, they form 'groups'.

Any channel that binds users together is a potential covert security breach. Since we are estimating the probability of intrusion, all of these must be considered. For example, a file, or set of files, connected to several users clearly forms a *system group*, in computer parlance. In graph-theory parlance the group is simply a *complete subgraph* or *clique*. In reality, there are many levels of association between users that could act as channels for communication:

- Group work association (access).
- Friends, family or other social association.
- Physical location of users.

In a recent security incident at a University in Norway, a cracker gained complete access to systems because all hosts had a common root password.

Each user naturally has a number of file objects that are private. These are represented by a single line from each user to a single object. Since all users have these, they can be taken for granted and removed from the diagram in order to emphasize the role of more special hubs (see fig. 2).

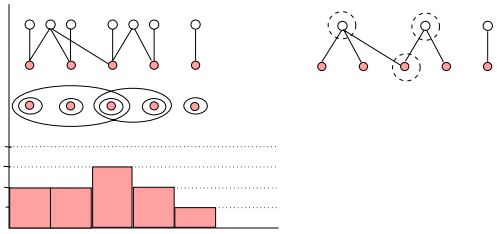

Figure 2: An example of the simplest level at which a graph may be reduced to a skeleton form and how hot-spots are identified. This is essentially a histogram, or 'height above sea-level' for the contour picture.

The resulting contour graph, formed by the Venn diagrams, is the first indication of potential hot-spots in the local graph topology. Later we can replace this with a better measure — the 'centrality' or 'well-connectedness' of each node in the graph.

4 Node centrality and the spread of information

In this section, we consider the *connected* components of networks and propose criteria for deciding which nodes are most likely to infect many other nodes, if they are compromised. We do this by examining the relative connectivity of graphs along multiple pathways.

Definition 1 (Degree of a node) *In a non-directed graph, the number of links connecting node i to all other nodes is called the degree k_i of the node.*

The best connected are the nodes that an attacker would like to identify, since they would lead to the greatest possible access, or spread of damage. Similarly, the security auditor would like to identify them and secure them, as far as possible. From the standpoint of security, then, important nodes in a network (files, users, or groups in the shorthand graph) are those that are 'well-connected'. Therefore we seek a precise working definition of 'well-connected', in order to use the idea as a tool for pin-pointing nodes of high security risk.

A simple starting definition of well-connected could be 'of high degree': that is, count the neighbours. We want however to embellish this simple definition in a way that looks beyond just nearest neighbours. To do this. we borrow an old idea from both common folklore and social network theory[13]: an important person is not just well endowed with connections, but is well endowed with connections to important persons.

The motivation for this definition is clear from the example in figure 3. It is clear from this figure that a definition of 'well-connected' that is relevant to the diffusion of information (harmful or otherwise) must look beyond first neighbours. In fact, we believe that the circular definition given above (important nodes have many important neighbours) is the best starting point for research on damage diffusion on networks.

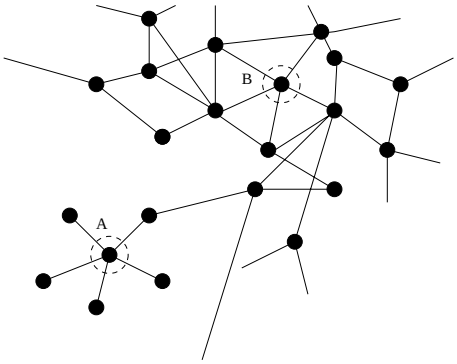

Figure 3: Nodes *A* and *B* are both connected by five links to the rest of the graph, but node *B* is clearly more important to security because its neighbours are also well connected.

Now we make this circular definition precise. Let v_i denote a vector for the importance ranking, or connectedness, of each node i. Then, the importance of node i is proportional to the sum of the importances of all of i's nearest neighbours:

$$v_i \propto \sum_{j=\text{neighbours of } i} v_j \quad . \tag{1}$$

This may be written as

$$v_i \propto \sum_j A_{ij} v_j \ , \tag{2}$$

where A is the *adjacency matrix*, whose entries A_{ij} are 1 if i is a neighbour of j, and 0 otherwise. Notice that this self-consistent equation is scale invariant; we can multiply $\vec{v}$ by any constant and the equation remains the same. We can thus rewrite eqn. (2) as

$$A\vec{v} = \lambda\vec{v}, \tag{3}$$

and, if non-negative solutions exist, they solve the self-consistent sum; i.e. the importance vector is hence an eigenvector of the adjacency matrix A. If A is an $N \times N$ matrix, it has N eigenvectors (one for each node in the network), and correspondingly many eigenvalues. The eigenvector of interest is the principal eigenvector, i.e. that with highest eigenvalue, since this is the only one that results from summing all of the possible pathways with a positive sign. The components of the principal eigenvector rank how 'central' a node is in the graph. Note that only ratios v_i/v_j of the components are meaningfully determined. This is because the lengths $v^i v_i$ of the eigenvectors are not determined by the eigenvector equation.

This form of well-connectedness is termed 'eigenvector centrality' [13] in the field of social network analysis, where several other definitions of centrality exist. For the remainder of the paper, we use the terms 'centrality' and 'eigenvector centrality' interchangeably.

We believe that nodes with high eigenvector centrality play a important role in the diffusion of information in a network. However, we know of few studies (see ref. [14]) which test this idea quantitatively. We have proposed this measure of centrality as a diagnostic instrument for identifying the best connected nodes in networks of users and files[2, 15].

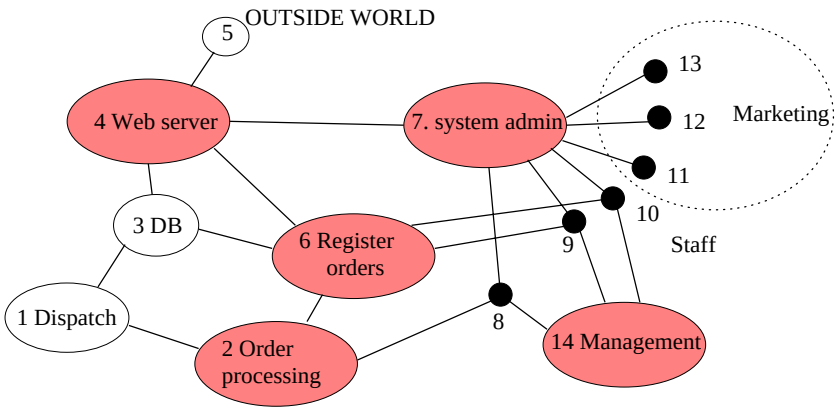

Figure 4: Unstructured graph of a human-computer system — an organization that deals with Internet orders and dispatches goods by post.

When a node has high eigenvector centrality (EVC), it *and its neighborhood* have high connectivity. Thus in an important sense EVC scores represent neighborhoods as much as individual nodes. We then want to use these scores to define clusterings of nodes, with

as little arbitrariness as possible. (Note that these clusterings are not the same as user groups–although such groups are unlikely to be split up by our clustering approach.)

To do this, we define as Centres those nodes whose EVC is higher than any of their neighbors' scores (local maxima). Clearly these Centres are important in the flow of information on the network. We also associate a Region (subset of nodes) with each Centre. These Regions are the clusters that we seek. We find that more than one rule may be reasonably defined to assign nodes to Regions; the results differ in detail, but not qualitatively. One simple rule is to use distance (in hops) as the criterion: a node belongs to a given Centre (ie, to its Region) if it is closest (in number of hops) to that Centre. With this rule, some nodes will belong to multiple regions, as they are equidistant from two or more Centres. This set of nodes defines the Border set.

The picture we get is of one or several regions of the graph which are well-connected clusters–as signalled by their including a local maximum of the EVC. The Border then defines the boundaries between these regions. This procedure thus offers a way of coarse-graining a large graph. This procedure is distinct from that used to obtain the shorthand graph; the two types of coarse-graining may be used separately, or in combination.

5 Centrality examples

To illustrate this idea, consider a human-computer system for Internet commerce depicted in fig. 4. This graph is a mixture of human and computer elements: departments and servers. We represent the outside world by a single outgoing or incoming link (node 5).

Let us find the central resource sinks in this organization, first assuming that all of the arcs are equally weighted, i.e. contribute about the same amount to the average flow through the organization. We construct the adjacency matrix, compute its principal eigenvector and organize the nodes into regions, as described above. The result is shown in fig. 5.

Node 7 is clearly the most central. This is the system administrator. This is perhaps a surprising result for an organization, but it is a common situation where many parts of an organization rely on basic support services to function, but at an unconscious level. This immediately suggests that system administration services are important to the organization and that resources should be given to this basic service. Node 6 is the next highest ranking node; this is the order registration department. Again, this is not particularly obvious from the diagram alone: it does not seem to be any more important than order processing. However, with hindsight, we can see that its importance arises because it has to liaise closely with all other departments.

Using the definitions of regions and bridges from section 4, we can redraw the graph using centrality to organize it. The result is shown in fig. 5. The structure revealed by graph centrality accurately reflects the structure of the organization: it is composed largely of two separate enterprises: marketing and order processing. These departments are bound together by certain bridges that include management and staff that liaise with the departments. Surprisingly, system administration services fall at the centre of the staff/marketing part of the organization. Again, this occurs because it is a critical dependency of this region of the system. Finally the web server is a bridge that connects both departments to the outside world — the outside hanging on at the periphery of the systems.

To illustrate the ideas further we present data from a large graph, namely, the Gnutella peer-to-peer file-sharing network, viewed in a snapshot taken November 13, 2001 [16]. In this snapshot the graph has two disconnected pieces—one with 992 nodes, and one with

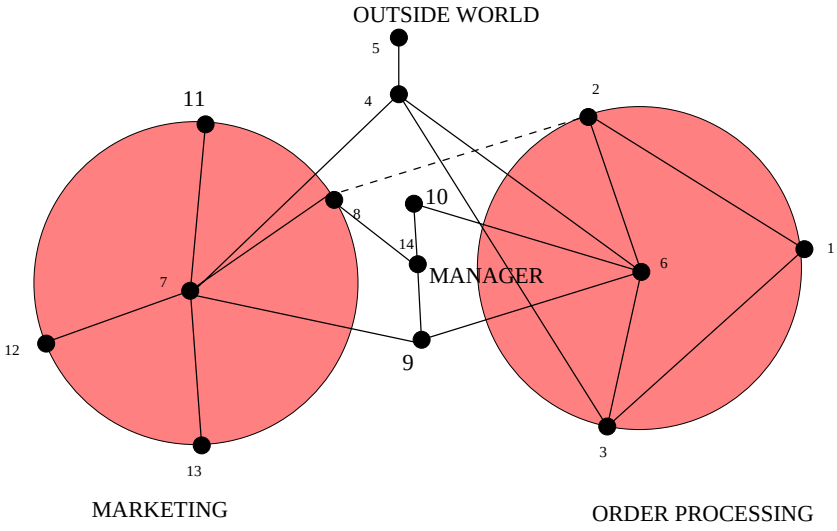

Figure 5: A centrality-organized graph showing the structure of the graph centred around two local maxima or 'most important' nodes, that are the order registration department and the system administrator. There are also 4 bridge nodes and a bridging link between the regions.

three nodes. Hence for all practical purposes we can ignore the small piece, and analyze the large one. Here we find that the Gnutella graph is very well-connected. There are only two Centres, hence only two natural clusters. These regions are roughly the same size (about 200 nodes each). This means, in turn, that there are many nodes (over 550!) in the Border.

In figure 6 we present a visualization of this graph, using Centres, Regions, and the Border as a way of organizing the placement of the nodes using our Archipelago tool[17].

Both the figure and the numerical results support our description of this graph as well-connected: it has only a small number of Regions, and there are many connections (both Border nodes, and links) between the Regions. We find these qualitative conclusions to hold for other Gnutella graphs that we have examined. Our criteria for a well-connected graph are consonant with another one, namely, that the graph has a power-law node degree distribution [10]. Power-law graphs are known to be well-connected in the sense that they remain connected even after the random removal of a significant fraction of the nodes. And in fact the (self-organized) Gnutella graph shown in fig. 6 has a power-law node degree distribution.

We believe that poorly-connected (but still percolating) graphs will be revealed, by our clustering approach, to have relatively many Centres and hence Regions, with relatively few nodes and links connecting these Regions.

6 Percolation: the spread of information in the graph

How many links or channels can one add to a graph, at random, before the system becomes essentially free of barriers? This question is known as the percolation problem and the breakdown of barriers is known as the formation of a *giant cluster* in the graph.

A graph is said to *percolate* if every node can reach every other by some route. This transition point is somewhat artificial for use as a management criterion, since links are constantly being made and broken, particularly in a mobile partially-connected

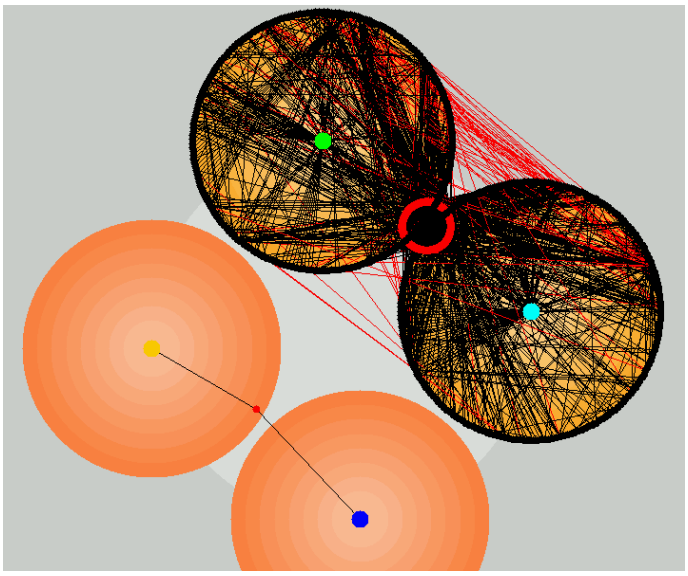

Figure 6: A top level, simplified representation of Gnutella peer to peer associations, organized around the largest centrality maxima. The graphs consists of two fragments, one with 992 nodes and one of merely 3 nodes and organizes the graoh into Regions. The upper connected fragment shows two regions connected by a ring of bridge nodes.

environment of modern networks. Rather we are interested in average properties and probabilities.

For small, fixed graphs there is often no problem in exploring the whole graph structure and obtaining an exact answer to this question. The most precise small-graph criterion for percolation comes from asking how many pairs of nodes, out of all possible pairs, can reach one another in a finite number of hops. The problem with these criteria is that one does not always have access to perfect information about real organizations. Even if such information were available, security administrators are not so much interested in what appears to be an accurate snapshot of the present, as in what is likely to happen in the near future.

To study a random graph, all we need is an estimate or knowledge of their degree distributions. Random graphs, with arbitrary node degree distributions p_k have been studied in ref. [12], using the method of generating functionals. This method uses a continuum approximation, using derivatives to evaluate probabilities, and hence it is completely accurate only in the continuum limit of very large number of nodes N.

We shall not reproduce here the argument of ref. [12] to derive the condition for the probable existence of a giant cluster, but simply quote it for a uni-partite random graph with degree distribution p_k.

Result 1 *The large-graph condition for the existence of a giant cluster (of infinite size) is simply*

$$\sum_k k(k-2)\ p_k \geq 0. \tag{4}$$

For a small graph with N nodes the criterion for a giant cluster becomes inaccurate. Clusters do not grow to infinity, they can only grow to size N at the most, hence we must

be more precise and use a dimensionful scale rather than infinity as a reference point. The correction is not hard to identify; the threshold point can be taken to be as follows.

Result 2 *The small-graph condition for widespread percolation in a uni-partite graph of order N is:*

$$\langle k \rangle^2 + \sum_k k(k-2)\, p_k > \log(N). \tag{5}$$

This can be understood as follows. If a graph contains a giant component, it is of order N and the size of the next largest component is typically $O(\log N)$; thus, according to the theory of random graphs the margin for error in estimating a giant component is of order $\pm \log N$. The expression in eqn. (5) is not much more complex than the large-graph criterion. Moreover, all of our small-graph criteria retain thier validity in the limit of large N. Hence we expect these small-graph criteria to be the most reliable choice for testing percolation in small systems. This expectation is borne out in the examples below.

7 Archipelago

Our reference implementation of the above criteria for testing node vulnerability and information flow, is a Java application program, which we call Archipelago.

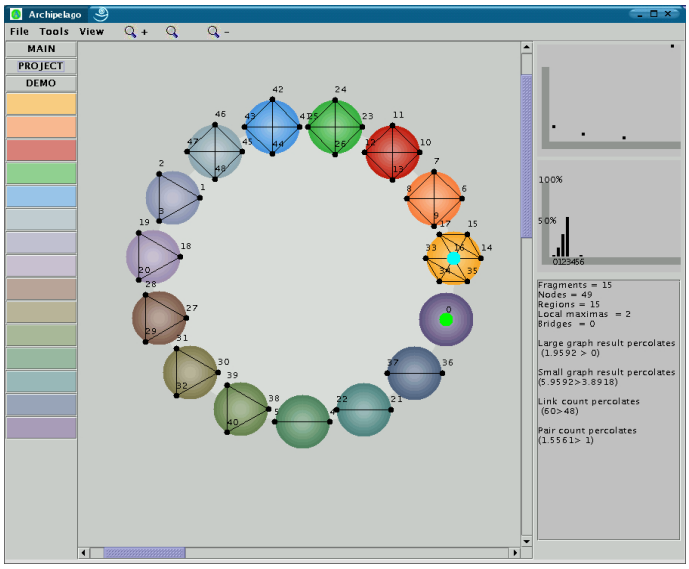

Figure 7: A scan of the student network at Oslo University College. This network is actually (in the absence of further links) quite secure against damage spreading, as it consists of many isolated pieces.

The Archipelago application accepts, as input, an adjacency matrix of a graph. This can be entered manually or generated, e.g. by a Perl script that scans file group associations. Archipelago calculates centrality and percolation criteria and organizes the regions into an archipelago of central peaks surrounded by their attendant nodes (coloured in black). Nodes and links that act as bridges between the regions are coloured red to highlight them, and disconnected fragments are coloured with different background tints to distinguish them (see figs. 7 and 8).

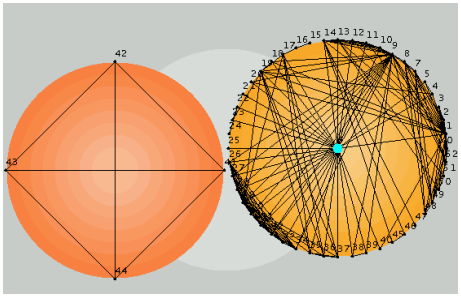

Figure 8: A scan of the staff network at Oslo University College. It is widely believed that this network is more secure than the student network, however this image shows otherwise. Since the staff are more trusting and more interconnected, the network is potentially far less secure.

A database of information about the nodes is kept by the program, so that regular SQL searches can be made to search for covert links between users, based on common properties such as same family name, or same address. The addition of even a single covert link can completely change the landscape of a graph and make it percolate, or depose unstable centres.

Analyses in the right hand panel of the main window (fig. 7) show the histogram of the degree distribution in the graph and a log-log plot of the same, in order to reveal power law distributions that are considered to be particularly robust.

8 Potential uses for Archipelago

We envisage several potential uses for this network analysis tool. We have already discussed some of these. Even armed with only centrality and percolation estimates, there is great flexibility in this mode of analysis.

- **Network robustness**: Determining how robust a network is to attack is an obvious task for the tool. Centrality determines the nodes that play the greatest role in the functioning of the system, and thus the obvious targets for attack. Techniques like these have been applied to the spread of viruses like HIV in the world of medicine. Peer to peer technology has been claimed to be extremely decentralized and therefore robust: there is no central control, and hence no obvious point of attack. One can use Archipelago to try 'taking out' these centres to see if the network can be broken up, and the spread of files curtailed. Attempting this has very little effect on the graph.

- **Resource investment**: In fig. 4, we considered how graphical analysis could be used to identify the places in a network where resources should be invested in order to maintain workflow. Here, a reorganization based on centrality illuminates the logical structure of the organization nicely. It consists of two regions: marketing and order processing, bound together by a human manager and a web server. The most central players in this web are the system administrator (who is relied upon by the staff and the servers), and the order processing department. The secure, continued functioning of this organization thus relies on sufficient resources being available to these two pieces of the puzzle. We see also an economic interpretation

to the system that speaks of continuity in the face of component failure. ISO17799 considers this to be a part of systemic security, and we shall not argue.

9 Conclusions

We have implemented a graphical analysis tool for probing security and vulnerability within a human-computer network. We have used a number of analytical tests derived in ref. [2]; these tests determine approximately when a threshold of free flow of information is reached, and localize the important nodes that underpin such flows.

At the start of this paper, we posed some basic questions that we can now answer.

1. *How do we identify weak spots in a system?* Eigenvalue centrality is the most revealing way of finding a system's vulnerable points. In order to find the true eigencentre of a system, one must be careful to include every kind of association between users, i.e. every channel of communication, in order to find the true centre.

2. *How does one determine when system security is in danger of breaking down?* We have provided two simple tests that can be applied to graphical representations. These tests reveal what the eye cannot necessarily see in a complex system, namely when its level of random connectivity is so great that information can percolate to almost any user by some route. These tests can easily be calculated. The appearance or existence of a giant cluster is not related to the number of groups, but rather to how they are interconnected.

An attacker could easily perform the same analyses as a security administrator and, with only a superficial knowledge of the system, still manage to find the weak points. An attacker might choose to attack a node that is close to a central hub, since this attracts less attention but has a high probability of total penetration, so knowing where these points are allows one to implement a suitable protection policy. It is clear that the degree of danger is a policy dependent issue: the level of acceptable risk is different for each organization. What we have found here is a way of comparing strategies, that would allow us to minimize the relative risk, regardless of policy. This could be used in a game-theoretical analysis as suggested in ref. [18].

Further ways of measuring centrality are being developed and might lead to new insights. Various improvements can be made to our software, and we shall continue to develop this into a practical and useful tool.

Availability

Archipelago is available from Oslo University College http://www.iu.hio.no/archipelago.

Acknowledgement

GC and KE were partially supported by the Future & Emerging Technologies unit of the European Commission through Project BISON (IST-2001-38923).

References

[1] M. Burgess. *Principles of Network and System Administration.* J. Wiley & Sons, Chichester, 2000.

[2] M. Burgess, G. Canright, and K. Engø. A graph theoretical model of computer security: from file access to social engineering. *Submitted to International Journal of Information Security*, 2003.

[3] L.E. Moser. Graph homomorphisms and the design of secure computer systems. *Proceedings of the Symposium on Security and Privacy, IEEE Computer Society*, page 89, 1987.

[4] J.C. Williams. A graph theoretic formulation of multilevel secure distributed systems: An overview. *Proceedings of the Symposium on Security and Privacy, IEEE Computer Society*, page 97, 1987.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *Computer Communications Review*, 29:251, 1999.

[6] A.L. Barabasi, R. Albert, and H. Jeong. Scale-free characteristics of random networks: topology of the world-wide web. *Physica A*, 281:69, 2000.

[7] A.L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[8] R. Albert, H. Jeong, and A.L. Barabasi. Diameter of the world-wide web. *Nature*, 401:130, 1999.

[9] B. Huberman and A. Adamic. Growth dynamics of the world-wide web. *Nature*, 401:131, 1999.

[10] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys*, 74, 2002.

[11] M.Y. Kao. Data security equals graph connectivity. *SIAM Journal on Discrete Mathematics*, 9:87, 1996.

[12] M. E. J. Newman, S.H. Strogatz, and D.J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:026118, 2001.

[13] P. Bonacich. Power and centrality: a family of measures. *American Journal of Sociology*, 92:1170–1182, 1987.

[14] G. Canright and Å. Weltzien. Multiplex structure of the communications network in a small working group. *Proceedings, International Sunbelt Social Network Conference XXIII, Cancun, Mexico*, 2003.

[15] G. Canright and K. Engø. A natural definition of clusters and roles in undirected graphs. *Paper in preparation*, 2003.

[16] Mihajlo Jovanovic. *Private communication*, 2001.

[17] M. Burgess, G. Canright, T. Hassel Stang, F. Pourbayat, K. Engø and Å. Weltzien. Automated security analysis. *Paper submitted to LISA2003*, 2003.

[18] M. Burgess. Theoretical system administration. *Proceedings of the Fourteenth Systems Administration Conference (LISA XIV) (USENIX Association: Berkeley, CA)*, page 1, 2000.