Cyber Security – The use of graphics in monitoring cyber security

The purpose of this learning module is to explore the types, use, and value of graphical interfaces in cyber security monitoring with a focus on 3D graphics. With the rapidly growing complexity of the internet and technology it is more and more critical to evaluate and identify cyber threats. This course provides an overview of the graphic monitoring available to display reports in a visual easy to read format versus a tabular text report that requires manual review to identify anomalies or critical data.

The course will review the process for taking a router log and developing potential cyber terrorist threat graphics by country, then demo an example of comparing a tabular router log report compared to a 3D graphical view of potential cyber terrorist's threat traffic.

The main areas of Cyber Security graphic visualization are the following[1]:

1. Intrusion Detection. Visualization of some network data in this area help researchers identify certain machines that try to gain unauthorized access to other machines.

2. Network Monitoring. In this area, visualization helps researchers identify unusually high traffic and track down points in the network that create such big traffic. It is very useful in order to identify computers launching Denial of Service attacks.

3. Border Gateway Protocol (BGP). This protocol manages reachability between hosts in different autonomous systems, i.e., networks controlled by Internet Service Providers. Visualization of BGP traffic is very important to ensure secure routing in the Internet.

4. Access Control. Access control visualization helps users and researchers understand complex policies that access control mechanisms en force.

5. Privacy. Privacy Visualization displays flow of information in various systems and gives intuition towards solutions that maintain privacy and deal with leakage of secret information.

6. Protocol Visualization. Complex protocols can be visualized and give better intuition to computer security experts to develop more efficient protocols (see e.g., [24]).

7. Attack Graphs. Attack Graphs are used to represent attack paths that an adversary can exploit in order to compromise a system. Based on vulnerabilities of certain parts of a system, an adversary can exploit one vulnerability after the other and reach the desired goal. Efficient visualization of attack graphs helps computer security analysts identify and fix vulnerabilities.

[1] Roberto Tamassia, Bernardo Palazzi, and Charalampos Papamanthou, "Graph Drawing for Security Visualization"

Table 1.
A table presenting combination of different Graph Drawing Methods with various Security Visualization areas.

	FORCED DIRECTION	LAYERED DRAWING	BIPARTITE DRAWING	CIRCULAR	TREEMAP	3D
Intrusion Detection	**22**					**77**
Network Monitoring	**13,9,15**		**26,1,5**			
BGP	**20**			**20**		**19**
Access Control		**14**			**10**	
Privacy		**25**				
Protocol Visualization		**24**				
Attack Graphs		**22,18**				

Intrusion Detection (ID)

Intrusion Detection (ID) and Intrusion Prevention (IP) systems are an indispensable part of the information security infrastructure of every networking company or organization. Intrusion Detections Systems (IDS) have problems, such as false positives, operational issues in high-speed environments and the difficulty of detecting unknown threats. Intrusion Prevention Systems (IPS) are still in their infancy. There is a misconception in the market that intrusion detection and intrusion prevention are basically the same technology or that IDS systems are on the way out and IPS and firewalls are the wave of the future. In fact, IDS are far from becoming obsolete and both systems complement each other. Much of ID research has focused on improving the accuracy and operation of IDSs but surprisingly there has been very little research in supporting the security analysts' intrusion detection tasks. In this example, we will describe an ongoing surveillance prototype system which offers a visual aid to the security analyst by monitoring and exploring 3D graphs. The system offers a visual surveillance of the network activity on a web server for both normal and anomalous or malicious activity. Colors are used on the 3D graphics to indicate different categories of web attacks and the analyst has the ability to navigate into the web requests, of either normal or malicious traffic.

Supporting Intrusion Detection by Graph Clustering and Graph Drawing ([22])

In this example the modelers apply graph drawing and graph clustering in order to provide a solution for intrusion detection. The presented system consists of: (a) a packet collecting unit, (b) a graph construction and clustering unit, (c) a visualization module and event generation module. The modelers model the

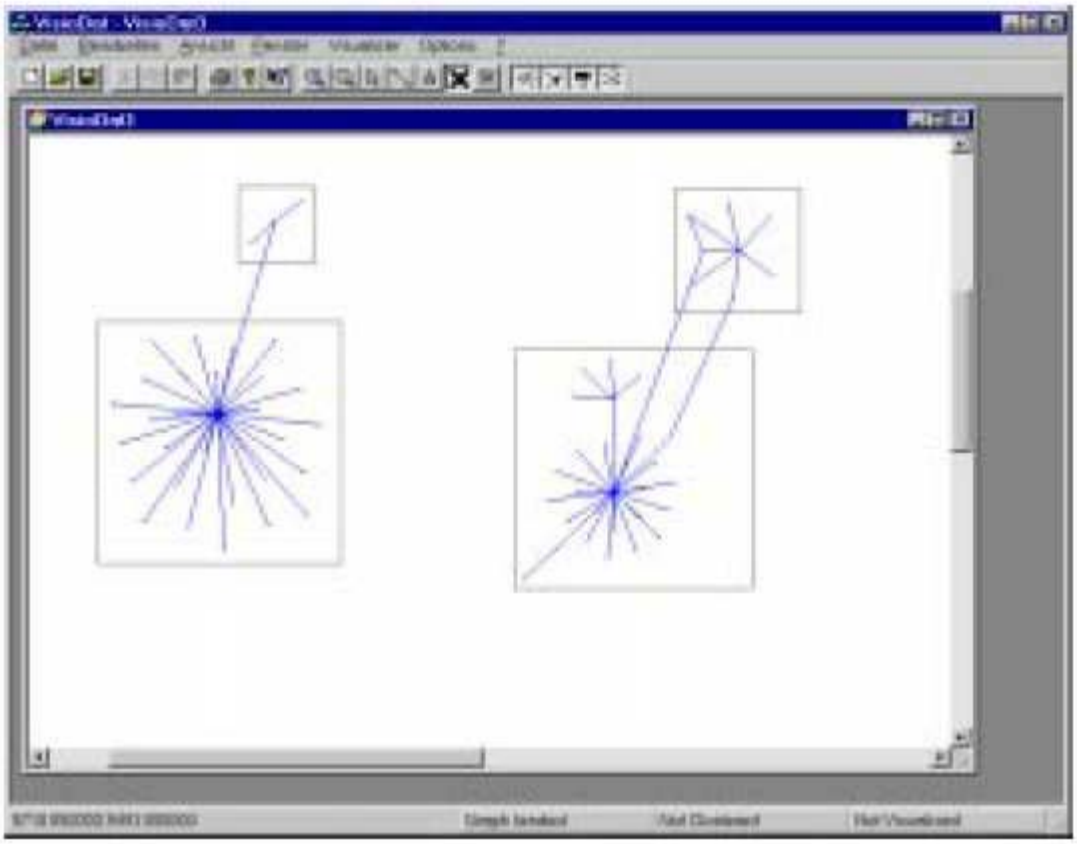

Fig. 1. Visualization of a network of computers using the proposed software for intrusion detection ([22]).

computer network with a graph where the nodes are the computers and the edges are weighted with the amount of exchanged data. Clustering is performed in the following way:

- Initially every node forms its own cluster.
- Then a node adopts the cluster where the majority of its neighbors belong to..
- Then the modelers use a spring embedder algorithm (e.g., [6], [8], [12]) to visualize the clusters and nodes within the clusters.

Note that forces are proportional to the weights of the edges, i.e., if there is a lot of communication between two nodes, these two nodes are closely placed. Also, in the visualization of clusters, an edge is placed between two clusters A, B if there is at least one edge between some node of cluster A and some node of cluster B.

The modelers propose that this tool can be used by the security analysts as follows. Every time they build the clustered graph, they keep some characteristics (feature vectors) of the graph (such as number of clusters, maximum degree of a node) and map these vectors to an intrusion detection method. Therefore a function from feature vectors to intrusion methods can be learned by means of regression.

Network Monitoring

Network Monitoring Visualization of Host Behavior for Network Security ([13])

In this example the modelers propose to visualize (over a certain time period) the different amounts and types of network traffic that come through a specific host using force-directed graph drawing techniques (network traffic visualization). Since there can be different types of traffic and there are also many time periods,

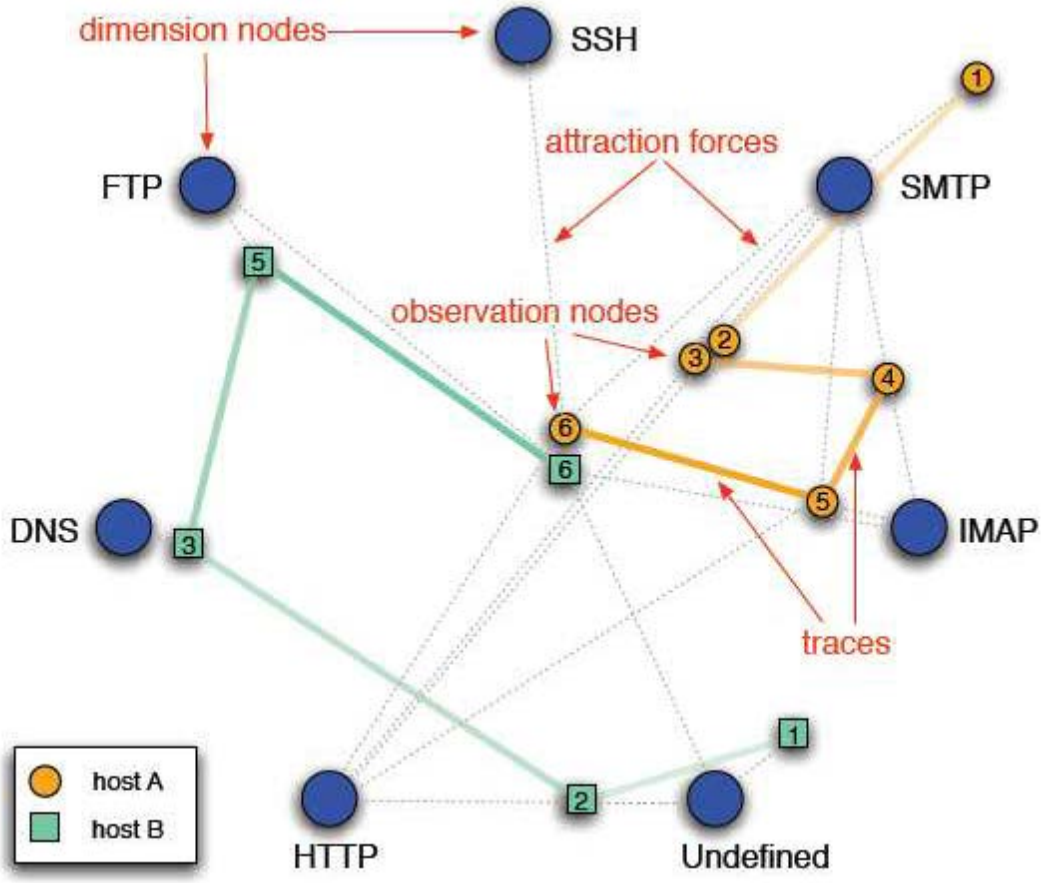

Fig. 2. Nodes: Dimension Nodes (types of traffic), Observation Nodes (time). Edges: There are edges between observation and dimension nodes and between different observation nodes ([13]).

the modelers try to turn this multi-dimensional problem into two dimensions so that they can use an efficient force-directed graph drawing method. The main "A Survey on Security Visualization Using Graph Drawing 5" nodes they use to set up the graph they visualize are of two types, the dimension nodes which are nodes that represent the types of visualized traffic, e.g., SSH, HTTP, FTP and also the observation

nodes which are nodes that represent the state of a certain host for a given time interval. For example, if the visualization refers to a period of 6 time intervals, there are going to be 6 observation nodes for every host. Also, there are virtual springs between observation and dimension nodes and also between observation nodes of the same host.
The modelers use a simple spring-embedder algorithm where there are attraction forces between observation nodes and dimension nodes.Also, there are repulsive forces between all the nodes. The modelers choose to use a modified version of the Fucheterman-Reingold algorithm to compute the layout [8], where they weights are used at the edges.

A Visual Approach for Monitoring Logs ([9])

In this example, the modelers propose an approach to visualize log entries that are obtained by monitoring network traffic. The log entries are basically k-dimensional vectors where each element of the vector corresponds to some characteristic of the network traffic, such as origin IP, destination IP, amount of exchanged traffic and others. The modelers wish to represent the entries of the log in the 2D plane by using graphs. To do that, they define a similarity measure between different entries in the log. The similarity measure they describe is very simple. They define the distance between two different entries in the log to be the sum of the differences of the respective fields of the log (for continuous fields). In order

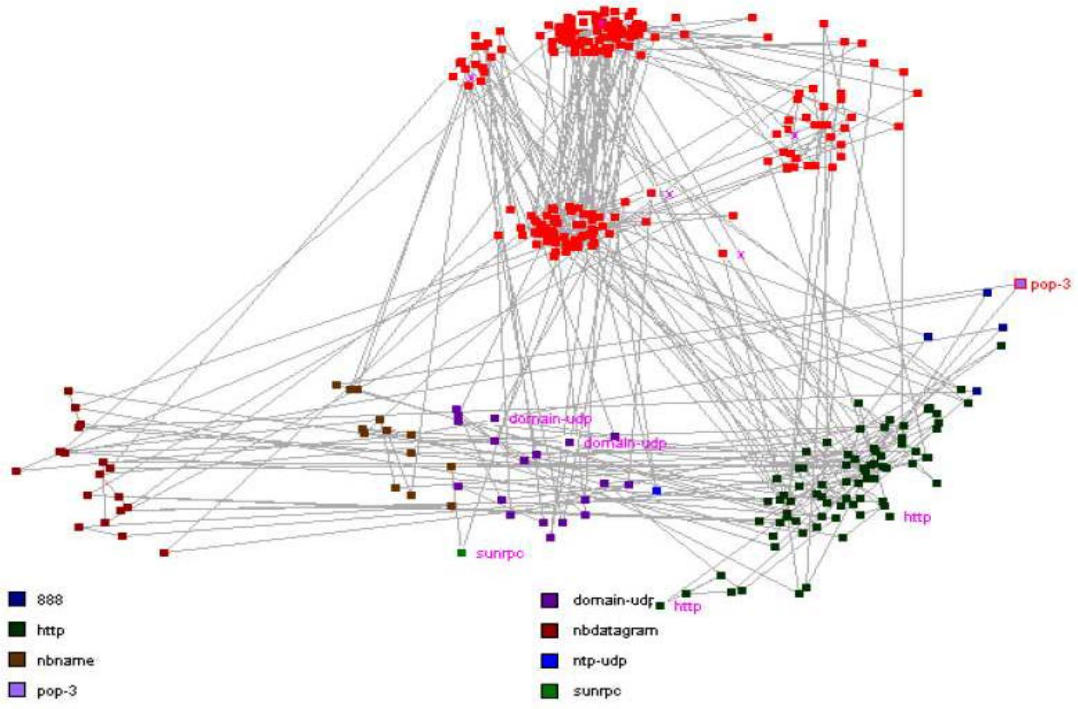

Fig. 3. Nodes are entries in the log. Also there is a spring between every two nodes and the power is proportional to the similarity ([9]).

to visualize the multidimensional data (entries of the log) in 2D, the modelers

use a force directed layout. They set up a graph where the nodes correspond to separate entries of the log and the forces between two nodes are proportional to the similarity of these two nodes. The modelers are using the algorithm from [4] to compute a 2D layout. This algorithm runs in linear time in the number of log entries and also the force between two nodes (log entries) i and j of the graph is proportional to dij − gij , where dij is the geometric distance between i and j whereas gij is their high-dimensional distance.

A Visualization Methodology for Characterization of Network Scans ([15])

In this example, the modelers develop a visualization system that shows the relationships between different network scans. The modelers set up a graph where each node represents a scan and the connection between them is weighted according to some metric (similarity measure) that is defined for the two scans. Some of Fig. 4.?? Every node is a scan. Weights on edges denote degree of similarity according to some metric. Edges exist if the weight is more than a certain threshold ([15]).

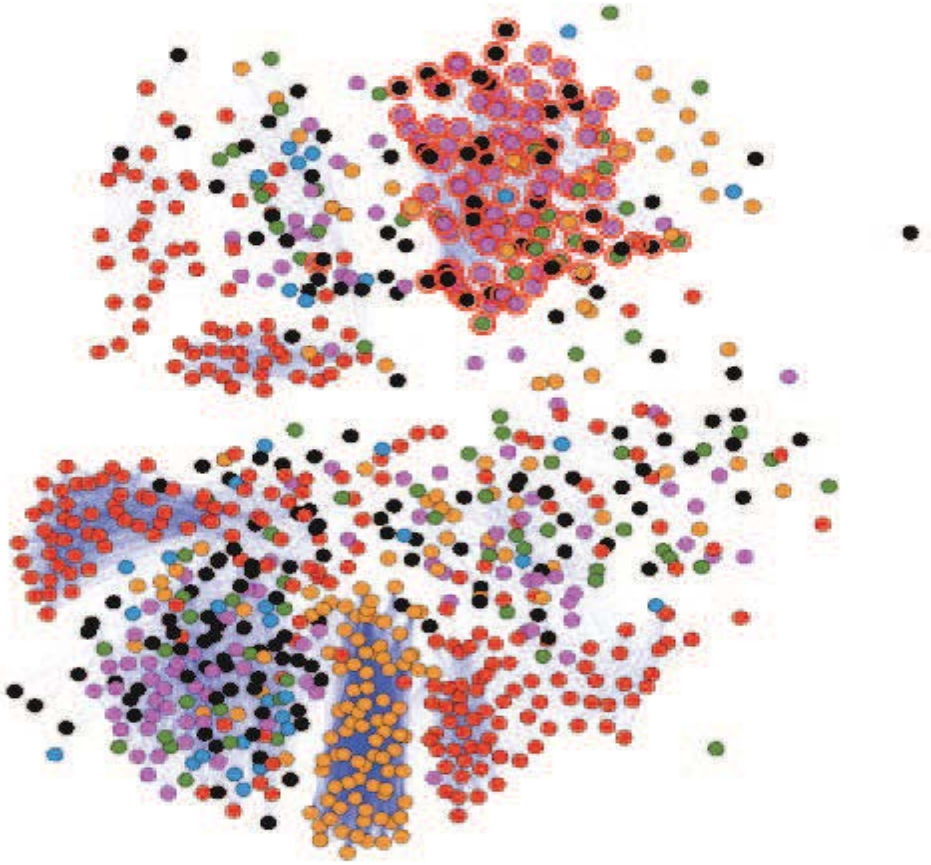

Fig. 4. Every node is a scan. Weights on edges denote degree of similarity according to some metric. Edges exist if the weight is more than a certain threshold ([15]).

The features taken into consideration for the definition of the similarity measure

are the origin IP, the destination IP and the time of the connection. Since the "A Survey on Security Visualization Using Graph Drawing 7" derived graph is a complete graph, in order not to have all the edges, the modelers define a threshhold and therefore every connection that has a weight less than a certain threshold is dropped. The modelers use the LinLog force directed layout [16] for the visualization. In the visualization produced, nodes with higher match percentages attract each other more than nodes that do not match as well.

Home-Centric Visualization of Network Traffic for Security Administration ([1])

In this example, the modelers use a matrix visualization technique combined with a simple graph drawing technique in order to visualize communication between domains in an internal network and domains in an external network. To visualize the internal IPs, the modelers use a square matrix. Every cell of the matrix corresponds to a certain IP of the internal network. Every cell is connected with an edge to other cells of variable area that represent the external IPs of our network. In this way we can identify which internal IPs have increasing

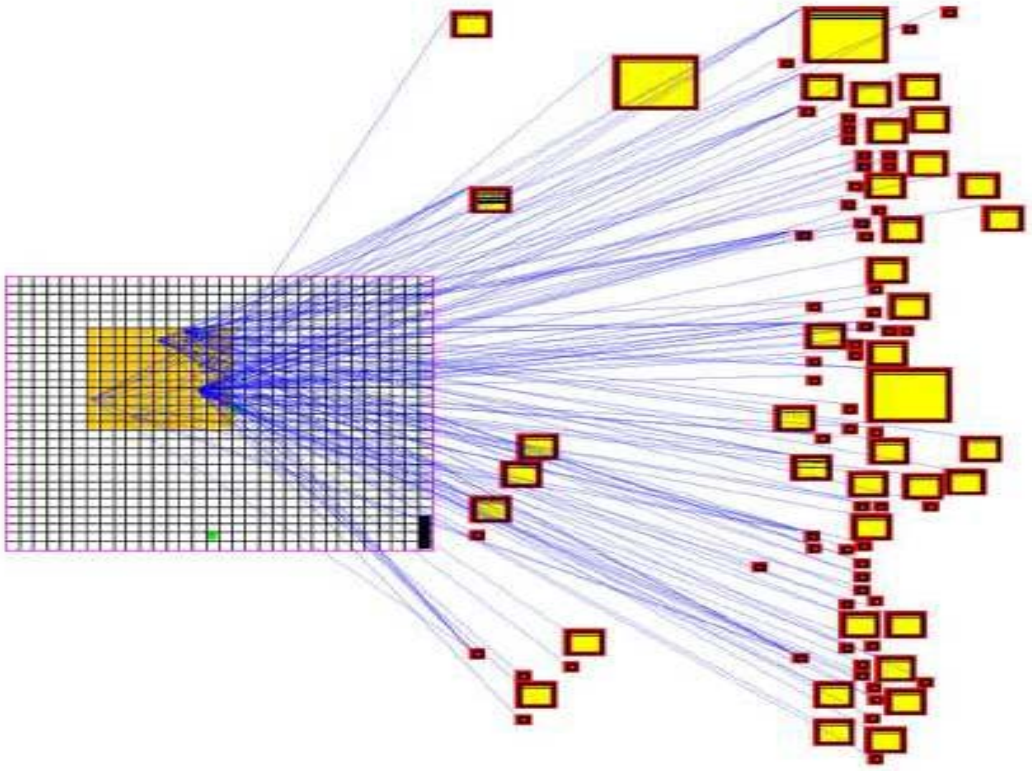

Fig. 6. Clear distinction between internal and external IP's. Internal IP's: the square matrix. External IP's: the yellow squares. The size of the yellow squares shows the relative amount of information exchanged between the specific host and the internal IP ([1]).

incoming or outgoing traffic and identify a virus outbreak, for example. Also we can

identify which external IPs were mostly hit from the internal IPs. The external IPs are visualized as variable size squares that lie in the area around the square matrix that represents the internal IPs. The size of each square is proportional to the number of packets sent/received by the specific external IPs. Therefore, squares of a large area correspond to external IPs of increased traffic which are likely to have launched an attack to some internal domains. Also the size of the squares of the external IPs shows relative amounts of activity among external IPs and therefore a comparison of different traffics between external IPs is feasible. Note that the existence of an edge between an internal host and an external host shows only that there existed some communication between the specific hosts. The exact amount of communication (number of packets) is represented with the size of the square of the external host. Finally note that the ordering of the internal IPs does not follow a specific ordering whereas the placement of the external IPs squares makes sure that overlaps are avoided.

Rumint: A graphical network monitor ([5])

Rumint is a free tool available at http://www.rumint.org/. The tool takes captured traffic as input and visualizes it in various unconventional ways. The most interesting visualization related to graph drawing is the parallel plot that allows one to see at one glance how multiple packet fields are related. The VCRcontrols functionality is interesting and allows for the ability to analyze different trends over time.

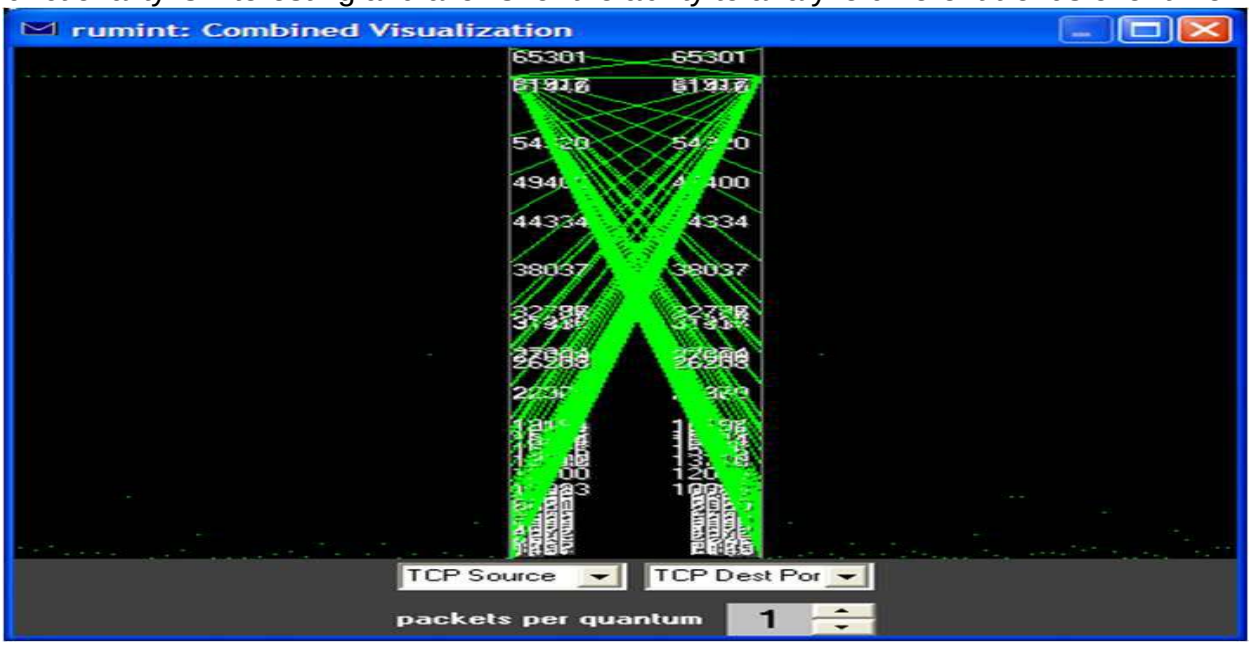

Fig. 8. Rumint ([5]) visualization of a *NMAP* scan using parallel plot, each dot indicate a different packet for the port on its row.

Border Gateway Protocol

VAST: Visualizing autonomous system topology ([19])

In this example the modelers propose to use different visualization techniques to visualize the Internet's BGP (Border Gateway Protocol) AS (Autonomous System) topology. The goal of the VAST tool is to allow security researchers to extract relevant information quickly from raw routing datasets. VAST is a 3D graphical tool that uses quad-tree to show per AS information and octo-tree to represent relationships between multiple AS. VAST tool main purpose is to visualize routing anomalies and sensitive points. In particular after a quick analysis on raw data it is possible to understand:

- a route leakage event, i.e., an anomaly in which an AS starts to re-announce a large numbers of prefixes into the Internet. Some portions of it are unreachable for the duration of the anomaly;
- a critical Internet infrastructure using historical BGP data
- portions of the Internet AS topology that are most important for its operation. For instance the modelers say that is easy to individuate the 10 most connected AS numbers in the Internet;

space hijacking incident where it is possible that an AS announces prefixes that do not belong to its domain caused by a wrong configuration.

In this case the tool helps to visually identify the anomaly very quickly.

The modelers developed another project, called Flamingo, that uses the same graphical engine as VAST but is used for real-time visualization of network traffic 4.2 BGP eye: A new visualization tool for real-time detection and analysis of BGP anomalies ([20]) In this example the modelers propose to use a new visualization tool, called BGP Eye, that provides a real-time status of BGP activity with easy-to-read layouts. BGP eye is a tool for visualization-aided root-cause analysis of BGP anomalies. The main objective of BGP Eye is to track the healthiness of BGP activity, raise an alert when an anomaly is detected, and indicate its most probable cause. BGP Eye allows two different types of BGP Dynamics visualization: Internet-Centric View and Home-Centric View, see Fig 11. Internet-Centric View studies the activity among ASes in terms of BGP events exchanged. In particular BGP Eye is used to visualize the Internet-AS network three different ways of laying out the network graph force-based layout, path distance based layout and manual based layout. Home-Centric View has been designed to understand the BGP behavior from the perspective of a specific AS, i.e., customer-AS. BGP updates originated and received by the customer-AS are clustered into different types of BGP events. The inner ring contains the routers of the customer-AS, the outer ring contains their peer routers, belonging to other ASes. In the outer layer, the method groups' routers belonging to the same AS are used as a node placement algorithm for the nodes to reduce the distance between connected nodes.

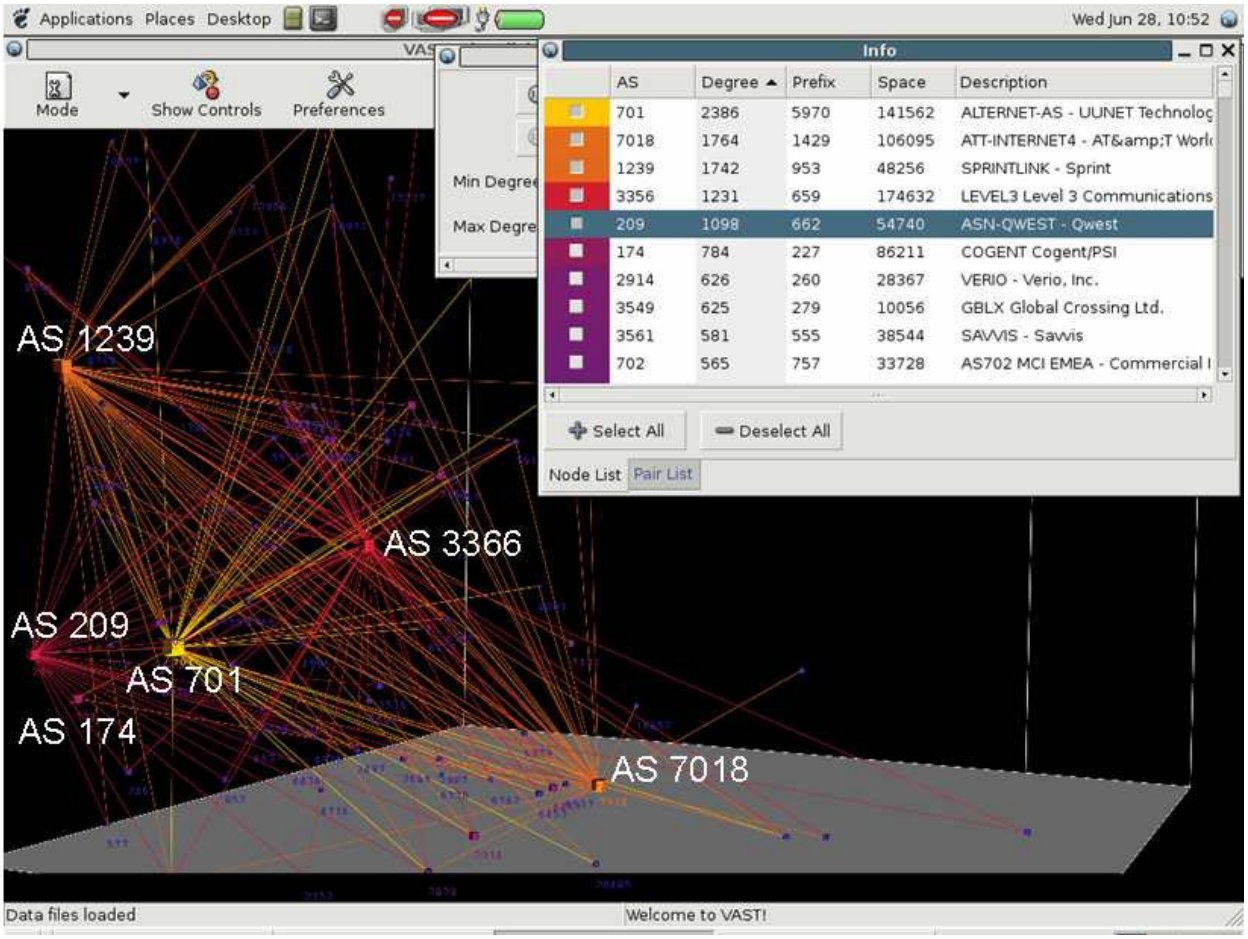

Fig. 9. VAST: A 3D graphical tool that visualizes the top Autonomous Systems in the Internet.[20]

Access Control

Information Visualization for Rule-based Resource Access Control ([14])

In this example the modelers provide a visualization solution that deals with another security problem, which is efficiently managing and querying rule-based resource access control. The modelers develop a tool that makes it easy to answer questions like "What group has access to which files during what time duration?" The development of tools that visualize rule-based access control systems is very important since the rules that define permissions in real-life systems tend to become more and more complex. In response, the modelers developed RubaViz, a visualization software that constructs a diagram for the administrator with subjects (people or processes), resources and groups of them. The software depicts connections from subjects to resources that are allowed from rules within policies. The modelers do not state something specificly about the visualization method they use to visualize the graph of the permission and rules.

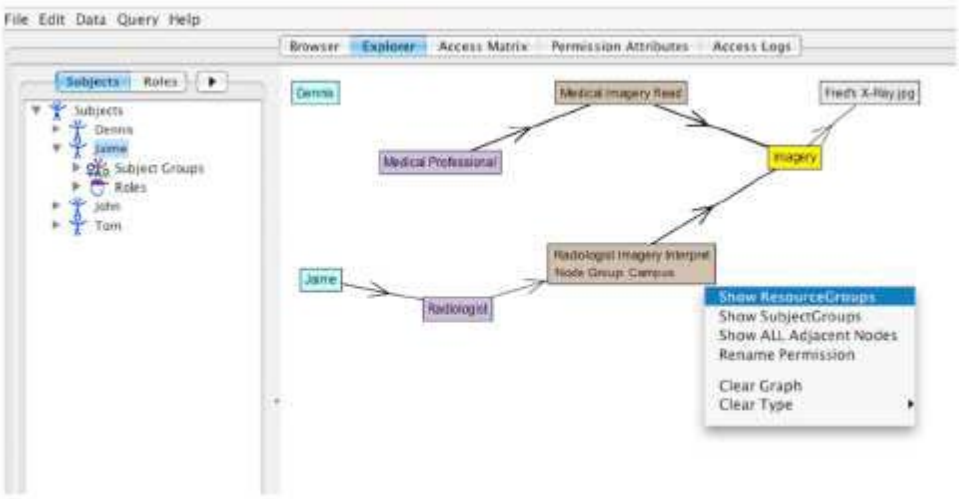

Fig. 12. A graphical explorer. Here, we see that Jaime, in the role of the Radiologist, has access to the Fred's X-Ray.jpg resource via the Radiologist Imagery Interpret rule ([14])

Effective Visualization of File System Access-Control ([10])

In this work, the modelers present a tool to visualize file permissions in the NTFS file system. It is very important for a user (or an administrator) to gain a global view of the different permissions of a file system because in this way he can resolve possible conflicts of permissions. Especially in the NTFS file system we have three kinds of different permissions, namely:
1. Explicit permissions which are set by the owner of each group/user;
2. Inherited permissions which are dynamically inherited from the explicit permissions of the ancestor folder;
3. Effective permissions which are obtained by combining the explicit and inherited permissions.

The tool presented in [10] uses a treemap layout to visualize the directory tree of the file system and also colors the tiles of the treemap with special colors to emphasize the relations between permissions of files and folders. In this way, it is made clear which permissions are a result of an "explicit" set or of a combination of an explicit set and the corresponding inherited permissions.

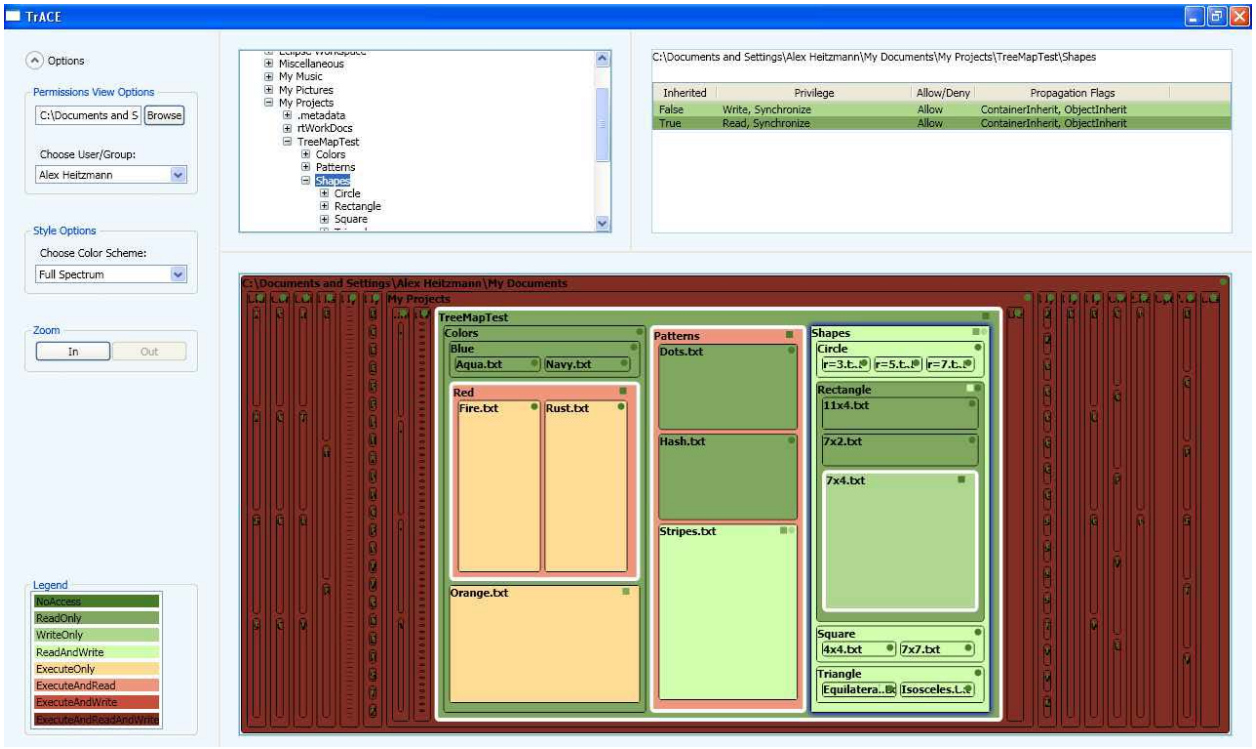

Fig. 13. Visualization of NTFS file system permissions using a treemap layout and different colors to display various permissions of files and folders ([10])

Privacy

Visualization for Privacy Compliance ([25])

In this example, the modelers propose a general visualization of diagrams showing the flow of private information for different applications. The proposed model tries to make it easier for the user to identify private information vulnerabilities that can lead to non-compliance. To achieve that goal the modelers develop a method of visualizing flowcharts where information flows are mapped out graphically so that one can see more easily the dangerous locations. The modelers present and analyze an example that refers to the flow of private information for an online pharmacy. The client sends his data to the online pharmacy and according to what the status of the drug repository is, the online pharmacy might have to send this information to other entities too so that the client can be served.
Therefore private information from the client is vulnerable. This is visualized with flowcharts and the modelers are able to detect some vulnerabilities. However. the modelers do not show or describe how their method can work for larger examples.

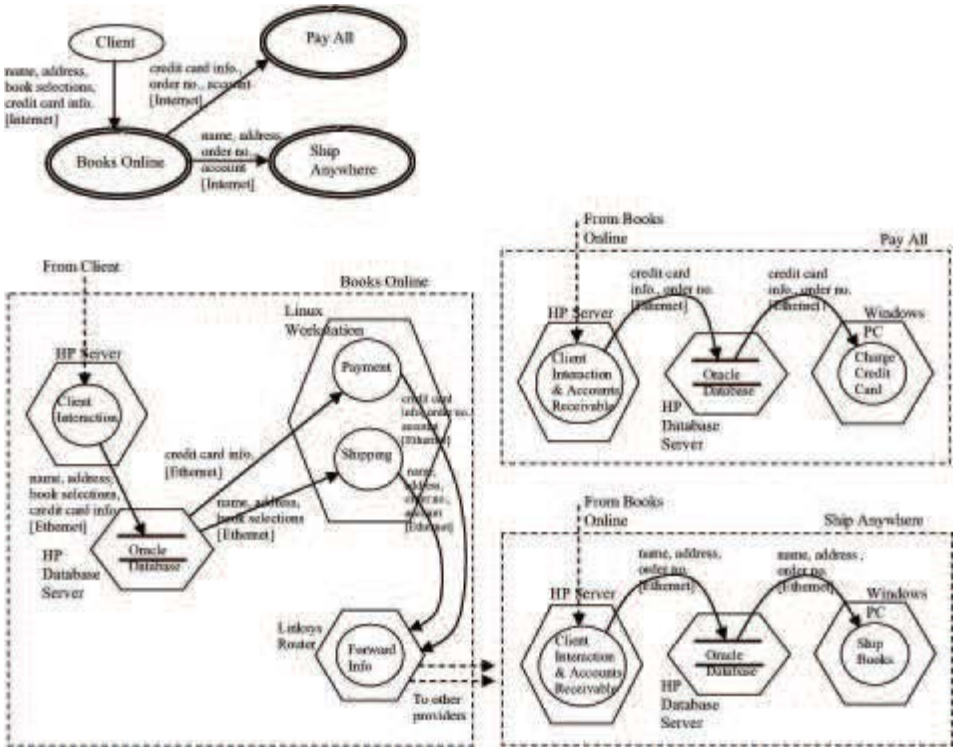

Fig. 14. Flowchart visualization for showing how private information flows through multiple transactions for an online bookstore ([25]).

Protocol Visualization

Visualization of Automated Trust Negotiation ([24])

In this example, the modelers propose to use graph drawing to visualize automated trust negotiation. This is another area of security visualization that is really interesting. The modelers claim that the visualization tool is provided with intuition for the improvement of the negotiation protocol. But what is automated trust negotiation (ATN)? ATN addresses the following web-service scenario: Suppose you do an online purchase. Most of the time you have to sign up for an account, that involves a lot of private information. (e.g., date of birth). This requirement can allow for private information leakage. ATN aims to solve this problem by designing negotiation protocols that operate on signed credentials. The ATN protocol can be represented with the trust-target graph (TTG) which is what is being visualized by the proposed tool.
In the trust-target graph protocol, a trust negotiation process involves the two negotiators working together to construct a trust-target graph (TTG). A TTG is a directed graph. Each node is either a trust target or a linking goal. The modelers use the Grappa system [2], a Java port of the Graphviz graph drawing system [7], to construct the drawings of the TTGs used in ATN negotiation sessions. As we can see the system uses layered drawing in order to visualize the TTG.

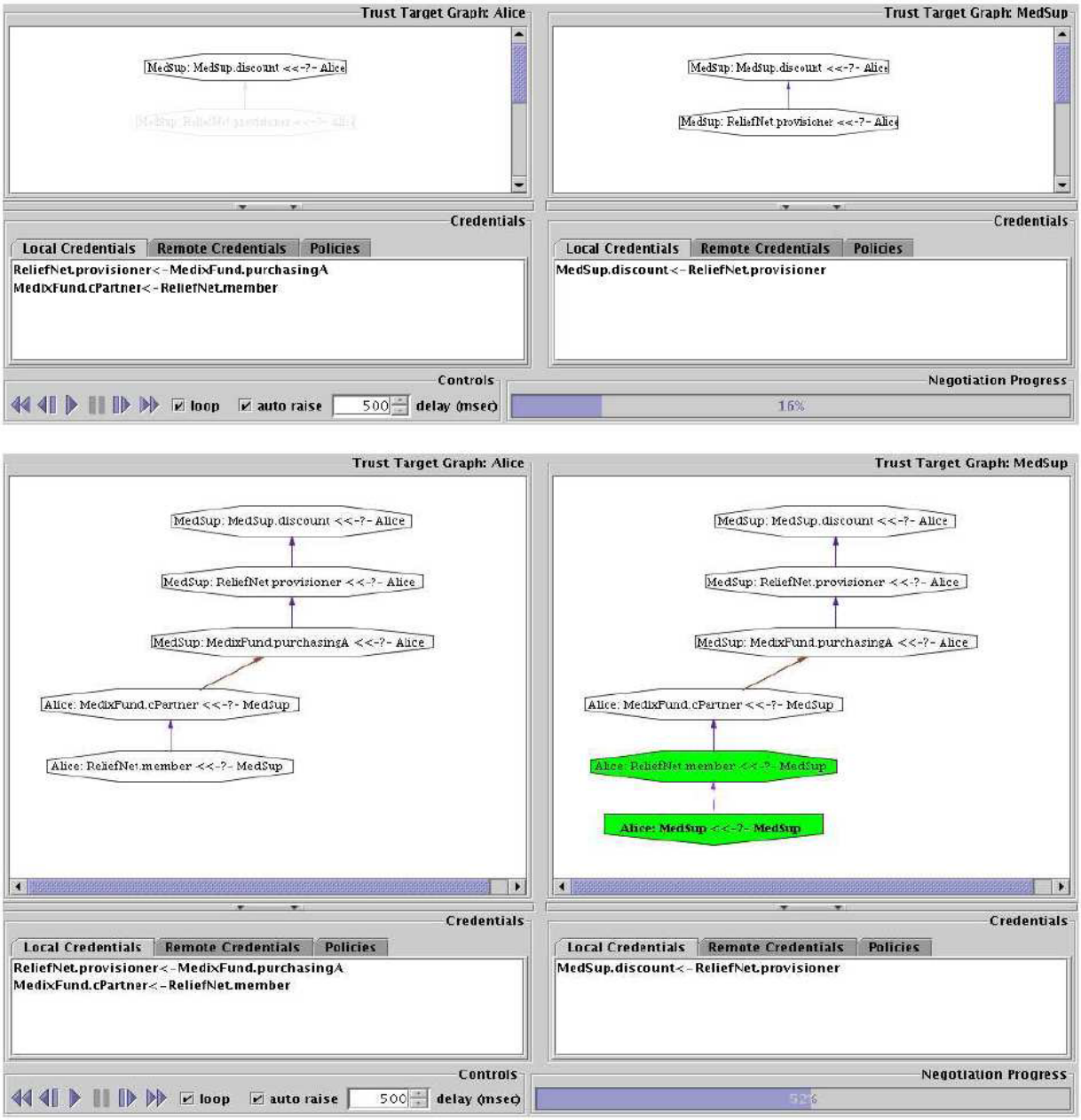

Fig. 15. Visualization of the trust-target Graph produced with a negotiation protocol ([24]).

Attack Graphs

Multiple Coordinated Views for Network Attack Graphs ([17])

In this example, the modelers visualize attack graphs using graph drawing techniques. Attack graphs are very important for computer security administrators and analysts. Given a system or a network and a database of known vulnerabilities that apply to certain parts of the system or the network, the computer security analysts can construct

graphs where each edge is an exploit of a certain vulnerability and allows the attacker to move from one machine (or subnet) to another.
However, the complexity of attack graphs can grow very easily and therefore the security analyst must create meaningful visualizations of them, so that countermeasures can be taken. Also, the modelers propose to use some kind of clustering

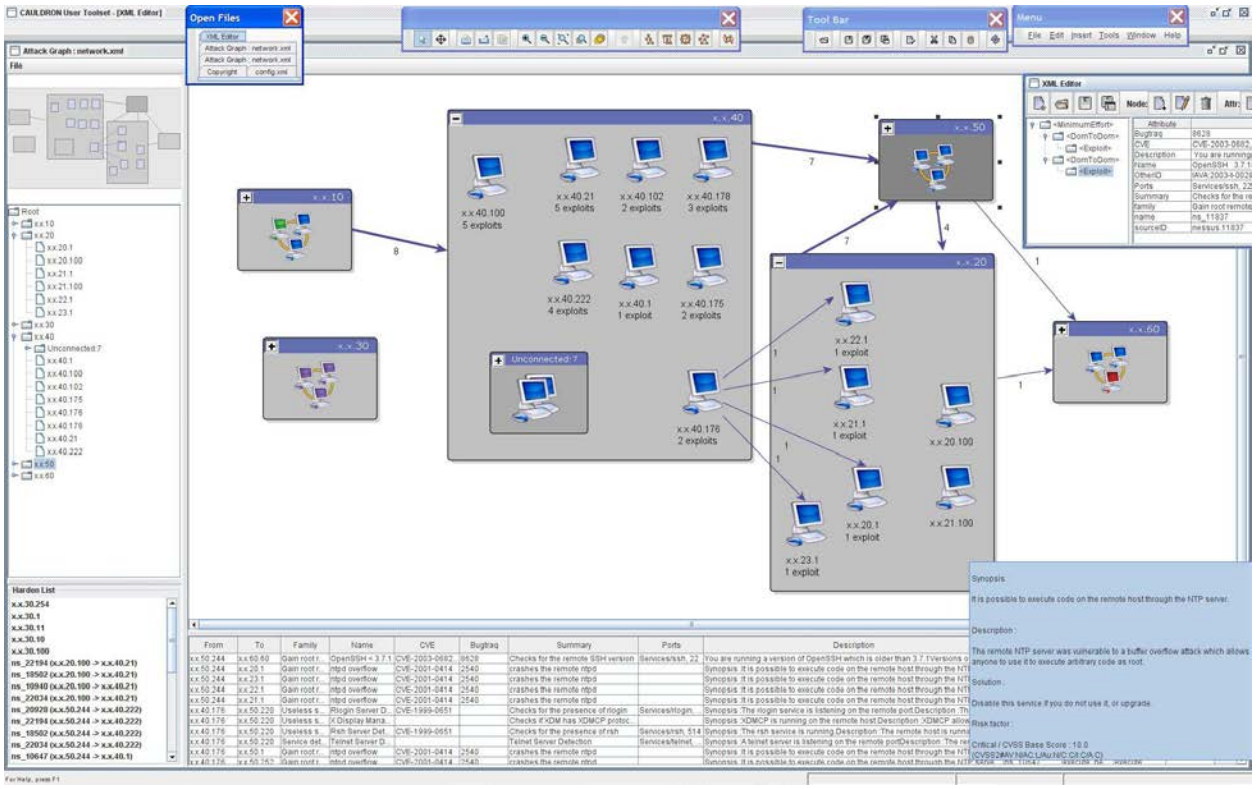

Fig. 16. An attack graph drawn with AT&T software Graphviz ([17])

of machines in order to reduce the complexity of attack graphs (e.g., machines that belong to the same subnet may be susceptible to the same attack and therefore they visualize the whole cluster as a node). In this example, the modelers use Graphviz [7] that produces a layered drawing of an attack graph. Also, similar layered drawings for attack graphs are proposed.[18].

3D Intrusion Detection Lab Example

This example of Intrusion Detection is performed using the Router, Telnet, Excel Spreadsheet, Visual Basic and the Nvidia 3D player on the Gateway System. The example process for taking a router log and developing potential cyber terrorist threat graphics by country then demo an example of comparing a tabular router log report compared to a 3D graphical view of potential cyber terrorist's threats traffic.

The following process is supported by Telnet, Router CLI, MS Excel spreadsheet using data manipulation, tables, and VisualBasic, and the display function of Nvidia 3D display.

1. The first step is to start the Telnet application and connect it to the router.
2. Logon to the router
3. Open the "Log" in Telnet
4. Save the log with a meaningful filename
5. Enter the following commands in the router: #show ip account, (Note: you must press the space bar until you get to the bottom of the log report)
6. Close the Log in Telnet
7. Enter following commands in router:

```
config t
#int port #'s e.g. ser1/1:0.1
#no ip account
Exit
Exit
Exit
```

8. Open the save log (Text file) in MS Excel
9. Navigate to Data tab and "Import", "Delimited", "Space", "Finis".
10. Reformat data so Source is in cell A1
11. Sort IP address by source
12. In the other worksheets have the Country IP ranges in tables.
13. Use Visual Basic search and compare country IP addresses
14. When there is a match add the packet total to that countrys Summary Report for the specific day.
15. When all IP addresses have been processed, graph the Summary Report
16. Save the Summary Report Graph in a left .png file
17. Save the Summary Report Graph in a right .png file
18. Reduce the size of the right .png file to 99% and save
19. Display the 3D file using the NVIDIA Visualization application.

The final report 3D graphical presentation is shown below [77]:

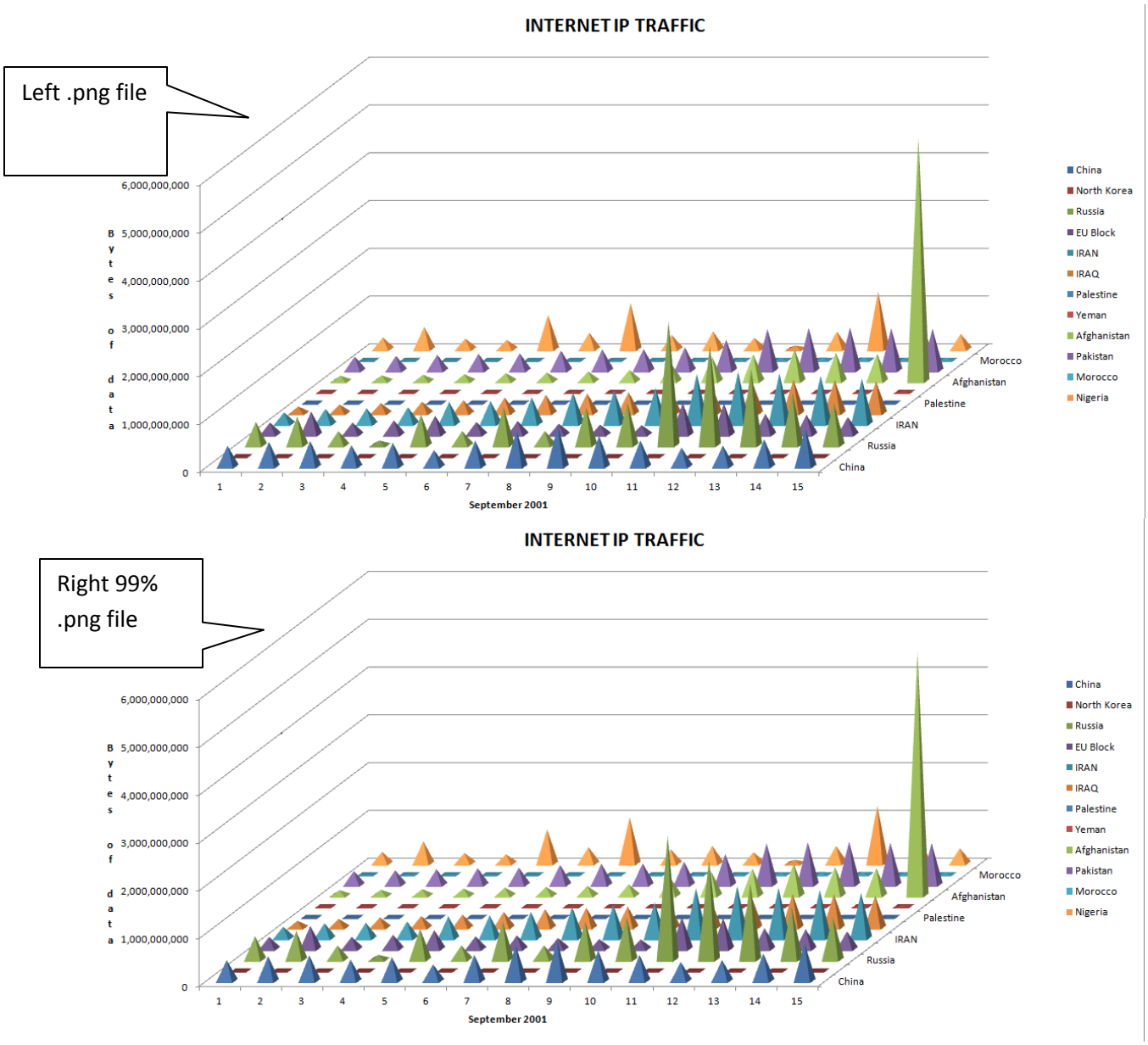

Figure xx 3D Intrusion detection graph [77] looking at IP addresses by country trying to access a router.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under grants IIS–0713403 and OCI–0724806, by the Kanellakis Fellowship at Brown University, and by the Italian Ministry of Research, grant number RBIP06BZW8, project FIRB “Advanced tracking system in intermodal freight transportation”.

References

[1] R. Ball, G. A. Fink, and C. North. Home-centric visualization of network traffic for security administration. In *VizSEC/DMSEC ’04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 55–64, New York, NY, USA, 2004. ACM.

A Survey on Security Visualization Using Graph Drawing 19

[2] N. S. Barghouti, J. Mocenigo, and W. Lee. *Grappa*: A GRAPh PAckage in Java. In G. Di Battista, editor, *Graph Drawing (Proc. GD ’97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 336–343. Springer-Verlag, 1997.

[3] G. D. Battista, F. Mariani, M. Patrignani, and M. Pizzonia. Bgplay: A system for visualizing the interdomain routing evolution. In *Graph Drawing*, pages 295–306, 2003.

[4] M. Chalmers. A linear iteration time layout algorithm for visualising highdimensional data. In *VIS ’96: Proceedings of the 7th conference on Visualization ’96*, pages 127–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[5] G. Conti. *Security Data Visualization*. No Starch Press, San Francisco, CA, USA, 2007.

[6] P. Eades. A heuristic for graph drawing. *Congr. Numer.*, 42:149–160, 1984.

[7] J. Ellson, E. R. Gansner, L. Koutsofios, S. C. North, and G. Woodhull. Graphviz and dynagraph - static and dynamic graph drawing tools. *Graph Drawing Software*, 2003.

[8] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Softw. – Pract. Exp.*, 21(11):1129–1164, 1991.

[9] L. Girardin and D. Brodbeck. A visual approach for monitoring logs. In *LISA ’98: Proceedings of the 12th USENIX conference on System administration*, pages 299–308, Berkeley, CA, USA, 1998. USENIX Association.

[10] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia. Effective visualization of file system access-control. In *Proc. Int. Workshop on Visualization for Cyber Security (VizSec)*, volume 5210 of *LNCS*, pages 18–25. Springer, 2008.

[11] B. Johnson and B. Shneiderman. Tree maps: A space-filling approach to the visualization of hierarchical information structures. In *Proc. IEEE Visualization*, pages 284–291, 1991.

[12] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inform. Process. Lett.*, 31:7–15, 1989.

[13] F. Mansmann, L. Meier, and D. Keim. Graph-based monitoring of host behavior for network security. In *Proc. Workshop on Visualization for Computer Security (VizSec)*, 2007.

[14] J. Montemayor, A. Freeman, J. Gersh, T. Llanso, and D. Patrone. Information visualization for rule-based resource access control. In *Proc. of Int. Symposium on Usable Privacy and Security (SOUPS)*, 2006.

[15] C. Muelder, K.-L. Ma, and T. Bartoletti. A visualization methodology for characterization of network scans. In *VIZSEC ’05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 4, Washington, DC, USA, 2005. IEEE Computer Society.

[16] A. Noack. An energy model for visual graph clustering. 2003.

[17] S. Noel, M. Jacobs, P. Kalapa, and S. Jajodia. Multiple coordinated views for network attack graphs. In *Proc. IEEE Workshop on Visualization for Computer Security (VizSEC)*, pages 99–106, 2005.

[18] S. Noel and S. Jajodia. Managing attack graph complexity through visual hierarchical aggregation. In *CCS Workshop on Visualization and Data Mining for Computer Security*, 2004.

[19] J. Oberheide, M. Karir, and D. Blazakis. Vast: visualizing autonomous system topology. In *VizSEC ’06: Proceedings of the 3rd international workshop on Visualization*

for computer security, pages 71–80, New York, NY, USA, 2006. ACM.
[20] S. T. Teoh, S. Ranjan, A. Nucci, and C.-N. Chuah. Bgp eye: a new visualization tool forreal-time detection and analysis of bgp anomalies. In *VizSEC '06: Pro*20 Roberto Tamassia et al.
ceedings of the 3rd international workshop on Visualization for computer security, pages 81–90, New York, NY, USA, 2006. ACM.
[22] J. T¨olle and O. Niggermann. Supporting intrusion detection by graph clustering and graph drawing. In *Proc. of Third International Workshop on Recent Advances in Intrusion Detection (RAID 2000)*, Toulouse, France, 2000.
[24] D. Yao, M. Shin, R. Tamassia, and W. H. Winsborough. Visualization of automated trust negotiation. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 8, Washington, DC, USA, 2005. IEEE Computer Society.
[25] G. Yee. Visualization for privacy compliance. In *VizSEC '06: Proceedings of the 3rd international workshop on Visualization for computer security*, pages 117–122, New York, NY, USA, 2006. ACM.