Crime analysis is a critical component of modern policing, and law enforcement agencies are increasingly using computerized analysis tools. A system developed at the University of Virginia adapts computerized techniques created for analyzing conventional crimes for use by law enforcement agencies in the Internet age.

Police departments and other law enforcement agencies continually analyze vast amounts of criminal incident data to better understand crime in their jurisdictions, to identify significant changes in crime levels, to plan community and neighborhood responses to crime, to investigate incidents, and to apprehend perpetrators. As the number of crimes increases and as resources—particularly human resources—for crime analysis remain relatively constant, law enforcement agencies are increasingly turning to automated tools to examine multiple incident reports and analyze criminal activities in their areas based on geographical location, time, and type.

The increase in computer crimes over the past decade requires enhancing the sophistication of crime analysis tools to address new types of crimes. According to Douglas Gansler, a state’s attorney in Montgomery County, Maryland, the range and extent of computer crimes is staggering, including “cyber-stalking, investment fraud, child sexual exploitation, information theft, domestic and international terrorism, copyright piracy, system deception, hate crimes, elder abuse, and plain old fraud.”

As the Internet grows in popularity and as e-commerce becomes a projected multi-trillion-dollar component of the economy, the number of computer crimes will rise accordingly. While regional crime analysis systems for conventional crimes are just now appearing, the Internet requires at least national and, ideally, global analysis tools. Further, “Internet time” requires crime analysis at faster rates and in significantly smaller time intervals than the days, weeks, and months typical of conventional crime analysis, and “Internet location” requires tools that are not fixed on geographical location.

Our computer crime analysis system builds directly on systems we researched and built for conventional crime analysis over large regions. These systems can link criminal activities by location, time, and method; they also can detect significant changes in criminal activity and discover criminal preferences to aid in predicting future threats.

**COMPUTER CRIME ANALYSIS**

To protect their systems’ continual operation and integrity, information systems operators support computer crime analysis through research efforts that include:

- broad-scope analysis of computer and network incidents and attacks to develop a system security policy as well as procedural and technological protection policies, and
- active system techniques to identify potential and real security vulnerabilities within specific systems.

Carnegie Mellon University’s Computer Emergency Response Team leads work in broad-scope crime analysis. Established in the late 1980s in response to the rapid evolution of the Internet and its attendant secu-
In its early period, CERT did only limited analysis of its incident and attack data. However, many researchers recognized a growing need to identify and classify computer and network vulnerabilities. In 1997, John Howard undertook a comprehensive review of CERT’s Internet security incident data. In this study, Howard reviewed the development of computer security incidents, developed a taxonomy of incidents, and provided security recommendations. This work provided a framework that analysts can extend to model and analyze other types of computer crime.

As both network and Internet usage increases, system intrusion detection grows in importance. A recent Carnegie Mellon technical report reviews current research and applications in this area, documenting the intrusion detection efforts and technologies that government, industry, and academia are using or considering. This work identifies a significant need for understanding an attacker’s motives because an intruder who, in previous attacks, was after military secrets will probably continue in this vein, while an intruder who was vandalizing an ex-employer’s database will probably continue along this line. Insights into the attacker’s goals would be invaluable since they provide a basis for action.

Researchers can use active data mining techniques to detect system intrusion. This method analyzes the behavior that intrusion causes and compares it to behavior that normal activities cause. Mining the raw audit data produces two types of information: association rules and frequent sequential patterns. Researchers use these results to construct an automatic classifier that distinguishes between invasive and normal behavior.

**ANALYSIS FRAMEWORK**

Our system is based mainly on the University of Virginia’s Regional Crime Analysis Program (Recap), which was created to provide a framework for analyzing crimes committed in different jurisdictions. Recap users can link related records, analyze trends in space and time, detect changes in those trends, and look for areas with a high density of criminal events—called hot spots.

Figure 1 shows Recap’s components—spatial analysis techniques that analysts implement using a geographical information system, statistical tools, and reporting tools. The geographical information system supports statistical analyses by geocoding information (locating incidents in space), computing spatial functions (for example, distances to roads and schools), and displaying results. The statistical tools allow change detection, incident report clustering or linking, and hot spot analysis and prediction.

Recap’s database supports multiagency analysis by drawing data from each agency or from multiple databases within a single agency and using national incident-based reporting standards and customized regional fields to define the incident fields. These customized fields include modus operandi (MO) information provided in narratives and not captured in other incident report fields.

Most computer crimes inherently have a multiagency nature because computer criminals, particularly those on the Internet, generally do not respect local police boundaries. Recap’s framework supports computer crime analysis by collecting and standardizing reports from individual agencies so that users can link crime reports over wide areas. The system also can link computer crimes to conventional crimes.

Recap’s statistical tools directly support the types of analysis needed for computer crimes. Of special interest are the data association tools, linking or clustering tools, and tools for analyzing current hot spots and forecasting threats. Analysts use these tools to link criminal activities by location, time, and method, detect significant changes in criminal activity, and understand individual criminal preferences so they can predict future criminal activity.

Recap’s spatial analysis component is also important for computer crime analysis, which does not necessarily use traditional maps. The topology of cyber-space crimes differs from the physical topography of most conventional crimes. This new “terrain” both restricts and defines the movements of criminals in ways that can significantly affect law enforcement.

In fact, spatial analysis tools may be more important for computer crimes than for conventional crimes.
In conventional crimes, law enforcement officers typically have both experiential and intuitive knowledge of the physical environment. However, the topology of the Internet and local networks present unfamiliar domains. Hence, mapping crimes within these topologies can help law enforcement authorities understand this relatively new realm of criminal activity.

CLUSTERING AND ASSOCIATING COMPUTER CRIMES

In the world of physical crime, crimes can be classified by type—mugging, murder, rape, and so on—and by the MO. Individuals tend to use a similar MO while committing different crimes, so the problem is one of clustering similar but not identical crimes.

Crime analysts work with incident reports containing all the information about a criminal event as recorded by a police officer. Analysts search these reports for indications of multiple incidents committed by the same person or group of persons. Once they identify this associated set of incidents, the analysts look for patterns in the way the criminal or gang operates and then use these patterns to aid in apprehension.

Data association

To automate this process, we used a data association methodology developed by Donald Brown and Stephen Hagen. This approach to associating records in databases depends on a measure of similarity or dissimilarity, which shows how closely two records match based on the values of individual attributes. Investigators use data association to cluster individuals or groups that tend to commit crimes with similar MOs, in similar areas, and at similar times.

Investigators use data association to determine that two crimes are similar—for example, one in which the intruder broke in the front door with a tire iron and another in which the intruder used a crowbar to break in the door. This technique also helps to determine that these crimes are different from one in which the criminal used a lock pick to open the door. Data association helps to define the limits of an investigation and provides insights into crime prevention measures. However, the volume of conventional crimes precludes manual data association in all but the most rare and important cases. For many common crimes—such as larceny—a single analyst would need an entire day to assess the associations between one new incident and 500 existing records. The problem is compounded for computer crimes, where the number of incidents is large and growing and where incident databases are spread over a larger number of agencies.

The data association methods we developed for analyzing conventional criminal incidents provide a resource for analysts facing the challenges of computer crimes. These association methods provide a similarity measure between different incidents. For example: Let \( \delta_k(A, B) \) denote the similarity of attribute \( k \) between criminal incident records \( A \) and \( B \), and let \( \omega_k \) denote the weighting of attribute \( k \). We compute the total similarity measure, \( TSM(A, B) \) between records \( A \) and \( B \) as a weighted sum of the attribute similarity measure \( \alpha \):

\[
TSM(A, B) = \frac{\sum_k \omega_k \delta_k(A, B)}{\sum_k \omega_k}
\]

However, this methodology requires that we define the relevant attributes, standardize the attribute functions in each agency database, define the similarity functions, and determine the weights. Experienced analysts define the relevant attributes collected in incident reports, then they standardize the values they obtain from different agencies. This standardization function typically involves conversion tables that integrate the concepts expressed within a single agency and between multiple agency databases.

For example, the crime of soliciting sex from a minor through a chat room can result in different entries in the incident reports of different agencies. Some may exclude the chat room name and simply report an Internet solicitation, while others may include the name. Still others may show the links to the chat room and initial contact information.

Concept hierarchies

Our approach uses concept hierarchies to link the values in different reports. For example, an interaction in Bob’s Sleazy Chat Room could be reported as occurring in a chat room or as an Internet interaction. However, if an interaction that occurs in a chat room is linked to an Internet interaction, we can calculate the similarity of these reports. We provide this kind of structure for each of the relevant attributes for associating computer crimes.

Next we need to define the association function \( \alpha(A, B) \). We use crime analysis data to develop mappings from the values in each report to a real number in the interval \([0,1]\). For example, let’s say that the attribute method of solicitation has three categorical values: e-mail, chat room, and carrier pigeon. Crime data analysis reveals that a solicitation in a chat room has a 0.7 similarity with a solicitation by e-mail and that a solicitation by carrier pigeon has a 0.001 similarity with either of the other two methods. In this manner, each categorical value for the relevant attributes in the reports is associated pairwise with the other values for that attribute.
Adjusting weight importance

Finally we calculate the weights for the attributes. Our approach allows the dynamic assignment of weights based on the incidents under investigation. To understand the need for dynamic weights, consider the following situation: Suppose we are investigating denial-of-service attacks. We might think the type of software used to generate the attack is important for linking incidents and give it a high weight in our total similarity measure. However, if every denial-of-service attack uses the same software, this attribute should get a low value because it is not important in discriminating between attacks.

The idea of adjusting weight importance has been widely explored in the multicriteria decision-making literature. In our approach, we first optimize the weights in the equation using cases that we know are associated. Essentially, we solve for the values of the weights—\( w_k \) in the equation above—that minimize the classification error, where this error is computed as the number of times we fail to join incidents that should be joined or times we join incidents that should not be joined. Then, for new incidents, we use an information theoretic factor—the relative entropy—to dynamically adjust the weights. This factor measures the expected information that the values in each report provide. The expectation is taken over the Bayesian posterior probability of association for the reports.

One of the most interesting aspects of computer crime is the amount of information available to criminals before they attempt the crime. But this also gives investigators a wealth of information for analyzing these crimes. Investigators use our data association methods to exploit the information available in large multiagency incident databases. For example, investigators can determine that two crimes are similar: one in which the thief stole a password and pasted obscene graffiti on a Web site and another in which a network packet sniffer captured a password and pasted obscene pictures on the site.

Investigators cluster the results of the total similarity measure to estimate the number of individuals or groups involved in cyberattacks. While many clustering methodologies could be used, we use the hierarchical agglomerative method with complete linkage. In this method

- if the similarity index of two entities is greater than a certain fixed value, or cutoff value, the entities form a cluster, and
- if one or both of the entities is a cluster, then all of the entities in both of the clusters have a similarity index that is greater than the cutoff value.

Investigators also use this algorithm to construct agents for multiagent simulation.

PREFERENCE DISCOVERY

A major goal of crime analysis is to understand the criminal processes at work in a region well enough to allow proactive policing activities. This means discovering areas and persons under threat and taking action to reduce the threat. Computer crime analysis has a similar goal.

There are many approaches to forecasting conventional crimes, ranging from time series methods to simulations. Our agent-based simulation approach uses preference discovery to develop a model of the criminal’s decision-making processes. We discover these criminal preferences in much the same way that Internet businesses discover customer preferences—by observing and analyzing behavior on the Web.

Figure 2 shows the basic components of this preference discovery approach. We observe criminal incidents in time and across the network topology, and we map these incidents into a feature space, which is defined by the relevant attributes of all incidents. For example, these attributes might include the characteristics of the site under attack, the type of tools used in the attack, and the time of day the attack was launched. We cluster these incidents in feature space. More formally, we develop a density estimate for the decision surface, which represents the criminal’s preference for specific attributes, across feature space. These surfaces then become the basis for modeling a criminal’s behavior in future attacks.

Our approach also employs a feature selection algorithm, which allows investigators to select the smallest

---

**Figure 2.** The criminal preference discovery approach, showing the relationship between the incident time, the incident location, and the clusters developed in the feature space. The time of the criminal incident is represented on the time axis, the location of the incident is represented on the network topology, and the clusters are represented in the feature space.
subset necessary to describe the criminal's observed preference. For example, we can distinguish between a computer criminal who attacks sites holding political views but is indifferent to economic profit made from these sites and a criminal who cares about the profit motive of a site but is indifferent to any political views. This suggests that in the first case, features that describe political views are relevant, while in the second case they are irrelevant. Because the result is the smallest possible feature set, we can use this information to determine the criminal's preference; we also can use it as an agent's preference in an agent-based simulation.

Our feature selection algorithm derives from our work on the analysis of conventional crime. Following this methodology, suppose that the incidents in the feature space shown in Figure 2 exhibit a clustering consisting of $C'$ clusters ($C'_i$: $i = 1, 2, \ldots, C'$). Now suppose that the incidents are truly related by a subset of the total number of attributes—for example, the political content of the site. It is then easy to show that if we use this subset for clustering, we will find at least as many clusters as we did with the complete set of attributes. Further, if there are $C$ criminals (or criminal groups) operating, we would expect to find no fewer than $C$ clusters. Hence, if we estimate $C$, we can selectively reduce the number of features until we find the "best" $C$ clusters. We determine the best clusters by calculating both within and between cluster distances.

We can also employ our data association approach prior to doing the preference discovery to find the preferences for individuals or groups.

**CRIME PREDICTION AND THREAT ASSESSMENT**

Figure 3 shows how all these pieces combine into one system. The database derived from multiple agency databases is the basis for the analysis. We employ data association to group incidents and use feature selection to select a set of features for the preference discovery. Preference discovery methods then generate the decision models for the criminal agents in our simulations. These models interact with additional Internet models that include guardianship and opportunity models. Guardianship models describe both the

---

*Figure 3. Agent-based criminal event prediction system. After data association determines the number of agents and feature selection determines their preferences, the model generates the agents. The features of importance to the agents determine the agents' environment. The model runs for a time step and generates a prediction as the simulation output. The model's tactical version predicts the behavior of computer criminals; its strategic version predicts how changes to the environment will change criminal behavior.*
Someone starting an investigation of a computer crime may face a massive amount of raw audit data. Here, automated crime analysis tools can be of great service, allowing the investigator to automatically sift through the data looking for nuggets of information. Data analysis will uncover a great deal of information, including both the number of criminals operating in the environment and their preferences, allowing the investigator to narrow the search to individuals or groups with those specific preferences.

For example, if a criminal's discovered preference is to disrupt sites with a specific set of features, the investigator could look for groups that have a reason to disrupt them. This knowledge considerably narrows down the scope of an investigation.

**Constructing a multiagent model**

To construct a multiagent model to simulate computer crimes, we derive the number and type of agents from the raw data audit, then we derive the criminal's preferences from the raw data. In this simulation, we use these derived preferences to create the criminal agents.

Researchers have used multiagent modeling to construct artificial societies in which computer models simulate social behavior. However, for this simulation we use the values derived from existing data to create a society that matches the one we actually observe.

**Linking and preference discovery**

Someone starting an investigation of a computer crime may face a massive amount of raw audit data. Here, automated crime analysis tools can be of great service, allowing the investigator to automatically sift through the data looking for nuggets of information. Data analysis will uncover a great deal of information, including both the number of criminals operating in the environment and their preferences, allowing the investigator to narrow the search to individuals or groups with those specific preferences.

For example, if a criminal's discovered preference is to disrupt sites with a specific set of features, the investigator could look for groups that have a reason to disrupt them. This knowledge considerably narrows down the scope of an investigation.

**Multiagent modeling**

Multiagent models provide an effective tool for predicting the behavior of computer criminals. These models use artificial agents, which can interact with their environment and with each other. While there is no universal definition of an agent, an agent is generally defined as having four properties:

- **Autonomy.** The agent does not need to be externally directed. Frequently, the agent has a set of rules to generate behaviors.
- **Reactivity.** The agent can perceive its surroundings and react to them.
- **Social ability.** The agent can interact with other agents in the model.
- **Proactivity.** The agent can initiate its own goal-directed behaviors.

Researchers have used multiagent modeling to construct artificial societies in which computer models simulate social behavior. However, for this simulation we use the values derived from existing data to create a society that matches the one we actually observe.
network environment. This allows a systems administrator to simulate an attack on the system to discover potential weak links. It also allows a policy maker to simulate the effects of changes in policies or procedures on a computer system’s security.

EXAMPLE ANALYSIS

Let’s say that the United Loomie Front (ULF), a separatist terrorist group, is fighting for the independence of the moon from the earth. We also have a group of apolitical hackers, Bob’s Pretty Good Hack Shop, which is in it strictly for the money. This particular incident starts when both a national journal and a New York paper publish articles supporting the earth’s continued political control over the moon.

The ULF launches two types of attacks:

• They commit a denial-of-service attack against the national journal and three of its advertisers (businesses 1, 2, and 3).
• They crack a password for the New York paper and write obscene graffiti all over the paper’s Web pages.

The hackers at Bob’s notice the denial-of-service attack and decide to use a distributed network sniffer to pull usernames and passwords from business 2 (a stock brokerage). Then, while the brokerage’s system administrators are busy keeping the system running, the hackers empty as many accounts as they can.

Figure 4 describes both the sources of the attack and the information that an outside observer will have about the attack. In keeping with the simplistic nature of this example, we have only included a small subset of all the possible sites on the Internet.

The feature space, in which the clustering will occur, has the following dimensions:

• Type of computer crime: damage versus fraud
• Type of attack: denial of service versus password cracking versus password sniffing
• Tool used: tool 1 versus tool 2 versus tool 3
• Type of business done at the site: jewelry versus stocks versus books versus news
• Type of sites that the site links to: commercial news versus retail versus noncommercial news

We assume that the experts have given us the following similarities between categorical values:

\[ \alpha_{(\text{damage, fraud})} = 0.10 \]
\[ \alpha_{(\text{denial of service, password cracking})} = 0.25 \]
\[ \alpha_{(\text{denial of service, password sniffing})} = 0.30 \]
\[ \alpha_{(\text{password cracking, password sniffing})} = 0.90 \]
\[ \alpha_{(\text{tool 1, tool 2})} = 0.75, \text{ and } \alpha_{(\text{tool 1, tool 3})} = 0.25, \]
\[ \alpha_{(\text{tool 2, tool 3})} = 0.25, \text{ and } \alpha_{(\text{sales, news})} = 0.75 \]
\[ \omega = 1 \]

Table 1 describes the total similarity matrix. If the cutoff for the clustering algorithm is set above 0.7, the algorithm will identify three groups (ABCD, E, F). If the cutoff for the clustering algorithm is set between 0.7 and 0.55, the algorithm will identify two groups (ABCDE, F). Although we need to determine the proper setting for the clustering algorithm, for this domain, the algorithm correctly identifies the attacks as coming from two separate groups.

The feature selection algorithm determines that the attackers have a preference for sites with the following characteristics:

• connected to either of the newspapers (ULF)
• either a newspaper site or a commercial site (ULF)
• a stock brokerage under another type of attack (Bob’s)

This is enough information to create a multiagent model to simulate a continued attack. In this simulation, we create two agents with the preferences derived above, and these agents act on the discovered preferences. Agent 1 (ULF) will continue to attack newspapers and commercial sites. It will use both denial of service and password cracking. Agent 2 (Bob’s) will continue to attack sites already under attack, that have a weakness allowing the use of a distributed network sniffer, and that have some type of account (brokerages, banks, credit card companies).

<table>
<thead>
<tr>
<th>Attack</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.9375</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.9375</td>
<td>0.9375</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0.9375</td>
<td>0.9375</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0.6875</td>
<td>0.6875</td>
<td>0.6875</td>
<td>0.6875</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>0.35</td>
<td>0.35</td>
<td>0.35</td>
<td>0.35</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>
Because these agents are acting on discovered preferences, we can predict their reaction to changes in the network environment. We can use this information to estimate the vulnerability of specific sites and perhaps to warn them. Figure 5 shows a map of the affected sites, with sites that are likely to be attacked shaded in magenta. Note that in this example, we have specified sites as either likely to be attacked or not likely to be attacked. In a real situation, we would give each site a probability of attack between 0 and 1.

Our crime analysis method uses data association to determine the number of criminal agents, then it uses feature selection to determine the preference of the identified agents. Because this method is automatable, analysts can use it in situations in which there is a vast wealth of data. Thus, this method is particularly useful in the computer crime domain, where data collection is relatively easy but data analysis is more difficult.

In addition, security personnel can use this method as the data source for a multiagent model that simulates future attacks without exposing new systems to the outside world.

Furthermore, analysts can use this method to study previous attacks, to estimate the number of attackers and determine their preferences, and to predict future attacks. This has applications in both the microdomain of an individual system—such as a local area network—and the macrodomain of an entire global system—such as the Internet or a private financial network. This methodology, then, will be of interest to members of both the computer and security communities.

Acknowledgments

Our work has been partially supported by grants from the Virginia Department of Criminal Justice Services and the National Institute of Justice, Crime Mapping Research Center.

References


**Donald E. Brown** is professor and chair of the Department of Systems Engineering at the University of Virginia. His research interests include threat analysis and system assurance. Brown received a PhD in industrial and operations engineering from the University of Michigan. He is past president of the IEEE Systems, Man, and Cybernetics Society. Contact him at deb@virginia.edu.

**Louise F. Gunderson** is a research assistant in the Department of Systems Engineering at the University of Virginia. Her research interests include modeling and simulation. She received an MS in environmental science from the University of Colorado. She is a member of INFORMS and ASPRS. Contact her at lfg4a@virginia.edu.

**Marc H. Evans** is a graduate student in the Department of Systems Engineering at the University of Virginia. His primary research interest is information assurance. He received a BA in international studies from the American University. He is a member of the IEEE, AFCEA, and AOC. Contact him at mhe8e@virginia.edu.

---

**Nine good reasons why close to 100,000 computing professionals join the IEEE Computer Society**

**Transactions on**

- **Computers**
- **Knowledge and Data Engineering**
- **Multimedia**
- **Networking**
- **Parallel and Distributed Systems**
- **Pattern Analysis and Machine Intelligence**
- **Software Engineering**
- **Very Large Scale Integration Systems**
- **Visualization and Computer Graphics**

[computer.org/publications/](computer.org/publications/)

---

**Sharpen your IT edge**

**Read about**

- Knowledge management programming for the Web
- e-commerce technologies
- protecting critical infrastructure networking
- information visualization

**In this year’s issues of IT Pro**

[computer.org/itpro/](computer.org/itpro/)