REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
05-07-2013	**Technical Report**	

4. TITLE AND SUBTITLE

The Diamond Model of Intrusion Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Sergio Caltagirone, Andrew Pendergast, and Christopher Betz

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

US Department of Defense
9800 Savage Rd.
Ft. Meade, MD 20755

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

US Department of Defense
9800 Savage Rd.
Ft. Meade, MD 20755

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper presents a novel model of intrusion analysis built by analysts, derived from years of experience, asking the simple question, "What is the underlying method to our work?" The model establishes the basic atomic element of any intrusion activity, the event, composed of four core features: adversary, infrastructure, capability, and victim. These features are edge-connected representing their underlying relationships and arranged in the shape of a diamond, giving the model its name: the Diamond Model. It further defines additional meta-features to support higher-level constructs such as linking events together into activity threads and further coalescing events and threads into activity groups. These elements, the event, thread, and group all contribute to a foundational and comprehensive model of intrusion activity built around analytic processes. It captures the essential concepts of intrusion analysis and adversary operations while allowing the model flexibility to expand and encompass new ideas and concepts. The model establishes, for the first time, a formal method applying scientific principles to intrusion analysis – particularly those of measurement, testability, and repeatability – providing a comprehensive method of activity documentation, synthesis, and correlation. This scientific approach and simplicity produces improvements in analytic effectiveness, efficiency, and accuracy. Ultimately, the model provides opportunities to integrate intelligence in real-time for network defense, automating correlation across events, classifying events with confidence into adversary campaigns, and forecasting adversary operations while planning and gaming mitigation strategies.

15. SUBJECT TERMS
cyber, cyber analysis, intrusion analysis, model, network defense, cyber threat, mitigation, computer network operations

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT UNCLASSIFIED	**b. ABSTRACT UNCLASSIFIED**	**c. THIS PAGE UNCLASSIFIED**	**SAR (Same As Report)**	**61**	**19b. TELEPHONE NUMBER** *(include area code)*

The Diamond Model of Intrusion Analysis

Sergio Caltagirone
sergio.caltagirone@cciatr.org

Andrew Pendergast
andrew.pendergast@cciatr.org

Christopher Betz
christopher.betz@cciatr.org

‘‘Intelligence analysts should be self-conscious about their reasoning process. They should think about how they make judgments and reach conclusions, not just about the judgments and conclusions themselves."

Richards J. Heuer Jr. [1]

‘‘Intrusion analysis is as much about tcpdump as astronomy is about telescopes" *Chris Sanders* [2]

Abstract

This paper presents a novel model of intrusion analysis built by analysts, derived from years of experience, asking the simple question, “What is the underlying method to our work?” The model establishes the basic atomic element of any intrusion activity, the event, composed of four core features: adversary, infrastructure, capability, and victim. These features are edge-connected representing their underlying relationships and arranged in the shape of a diamond, giving the model its name: the Diamond Model. It further defines additional meta-features to support higher-level constructs such as linking events together into activity threads and further coalescing events and threads into activity groups. These elements, the event, thread, and group all contribute to a foundational and comprehensive model of intrusion activity built around analytic processes. It captures the essential concepts of intrusion analysis and adversary operations while allowing the model flexibility to expand and encompass new ideas and concepts. The model establishes, for the first time, a formal method applying scientific principles to intrusion analysis – particularly those of measurement, testability, and repeatability – providing a comprehensive method of activity documentation, synthesis, and correlation. This scientific approach and simplicity produces improvements in analytic effectiveness, efficiency, and accuracy. Ultimately, the model provides opportunities to integrate intelligence in real-time for network defense, automating correlation across events, classifying events with confidence into adversary campaigns, and forecasting adversary operations while planning and gaming mitigation strategies.

Contents

List of Figures

1 Introduction

The discipline of intrusion analysis has existed since the discovery of the first intrusion.[1] External hackers and malicious insiders, mostly slyly, infiltrate and attack while intrusion analysts and system administrators work to uncover, understand, and thwart their operations. The questions remain little-changed since the discipline's epoch: the who, what, when, where, why, and how. Historically, these questions informed incident response to address the activity at-hand but defenders lacked the models and frameworks for activity documentation, synthesis, and correlation necessary to answer a question of growing importance: will the adversary return as part of a coordinated campaign? Yet the question ultimately leads organizations away from tactical mitigation (countering the activity) and towards strategic mitigation (countering the adversary) thereby increasing the effectiveness of mitigation and the adversary's cost to conduct operations.

This paper presents a novel model of intrusion analysis built by analysts, derived from years of experience, asking the simple question, "What is the underlying method of our work?" It arrives at its name, the Diamond Model, for its simple organization of the most fundamental aspects of malicious activity into the shape of a diamond. Our model establishes, for the first time, a formal method applying scientific principles to intrusion analysis: those of measurement, testability, and repeatability – providing a simple, formal, and comprehensive method of activity documentation, synthesis, and correlation. This scientific approach and simplicity produces improvements in analytic effectiveness, efficiency, and accuracy.

Our model is at once simple and complex, informal and formal, useful for analysis of both insider and external threats. Informally, analysts easily comprehend the model making it useful in the heat of pursuit on a daily basis. The model is the basis of an ontology[2] and presents a framework upon which to discover new activity, maximize analytic pivot opportunities, correlate and synthesize new information, and pursue the adversary over time, all while improving communication and documentation.

Formally, the model is a mathematical framework allowing the application of game, graph, and classification/clustering theory to improve analysis and decision making. The formality provides several benefits: testable analytic hypotheses ensuring repeatability and accuracy of analytic results, easier hypothesis generation, automated correlation across events, quickly classifying events with confidence into adversary campaigns, and forecasting adversary operations while planning and gaming mitigation strategies. Ultimately, this formality leads to the model's ability to integrate correlated intelligence for network defense capabilities, easily evolving to adopt new adversary infrastructure, capabilities, and processes.

Most importantly, the model is purposefully generic and thereby expandable and flexible. It accurately captures the essential concepts of intrusion analysis and adversary operations.

[1] In this paper the term "intrusion" is used to denote all malicious and nefarious activity targeting computer systems and networks.

[2] The model does not present a new ontology, taxonomy, sharing format, or protocol but by its fundamental nature should form the basis of these. This view is supported by others in [3].

These attributes enhance the model's utility, allowing it to grow and encompass new ideas and concepts.

2 Related Work

In intrusion analysis, we stand with analysts and experts such as Stoll [4], Bellovin [5], and Cheswick [6] who have been discovering and documenting malicious events with little-to-no formal training and tools. They have often relied on reams of data print-outs to analyze the activity and armed only with intuition and supreme technical ability. Their initial tradecraft documentation and story-telling led many analysts down the path of hunting adversaries. Modern intrusion analysts continue this tradition with outstanding and innovative efforts like the Honeynet Project [7].

Northcutt in [8] and others have strengthened analytic training by showcasing specific threat activity examples and providing students the opportunity to understand adversary tools and tradecraft. Hands-on analytic training from organizations such as SANS [9] is now a significant source of analytic tradecraft dissemination.

While these stories, papers, books, and courses provide solid cases for teaching the mechanics of intrusion analysis, they do not offer the scientific approach necessary to underpin the process. Without the underlying model (either formal or informal) to explain how analysts evaluate and understand malicious activity, efforts to evolve tradecraft are difficult to impossible.

Additional work tangentially relates to intrusion analysis and the use of intelligence-driven network defense. Amann, et al., in [10] accurately states, "it is becoming increasingly difficult to reliably report today's complex attacks without having external context at hand. Unfortunately, however today's IDS [intrusion detection systems] cannot readily integrate intelligence..." Their work significantly advances the ability for an IDS to integrate external context and threat intelligence in real-time to increase detection success. This is a critical capability for future mitigation which the Diamond Model complements by identifying how analysts effectively, efficiently, and accurately develop that external context and intelligence to enrich detection.

The 'Kill Chain' provides a highly effective and influential model of adversary operations which directly informs mitigation decisions [11]. Our model integrates their phased approach and complements Kill Chain analysis by broadening the perspective which provides needed granularity and the expression of complex relationships amongst intrusion activity. This allows the full scope of knowledge to be represented as opposed to only the observable indicators of the activity. Furthermore, our model provides a formal mathematical method for effective graph analysis and grouping (e.g., clustering/classification) to solve many classes of analytic problems. This feature allows the model to support numerous complementary strategy planning frameworks such as the Joint Intelligence Preparation of the Operational Environment (JIOPE) [12], course of action matricies [11], the Active Defense Algorithm

and Model (ADAM) [13], and potentially more "cutting-edge" strategy development using evolutionary computing techniques such as [14].

Traditional attack graphs attempt to generate all possible attack paths and vulnerabilities for a given set of protected resources to determine the most cost effective defense and the greatest degree of protection. Attack graphs originated with Schneier's "Attack Trees" and evolved into a valuable vulnerability analysis tool to develop effective defense-in-depth strategies [15]. Until 2005, attack graphs faced significant difficulties in scalability, measurement, and usability [16]. However, progress has been made improving scalability for real-sized networks [17, 18], measurement [19], and usability [20]. Our model defines a new intelligence-centric attack graph, called activity threads, and combines intelligence and traditional attack graphs into an activity-attack graph. Activity-attack graphs merge traditional vulnerability analysis with knowledge of adversary activity. They integrate what has occurred with the potential and preferred attack vectors enabling more effective analysis and mitigation strategy development. This ultimately allows a more efficient allocation of defense resources. Additionally, prior work in [21] has already shown the applicability of attack graphs directly in intrusion detection systems. This allows the activity threads and adversary processes developed in our model to be directly implemented in intrusion detection systems.

Many systems, languages, and taxonomies have been developed which allow analysts to document malicious activities and share indicators [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. Our model does not propose an ontology, taxonomy, or sharing protocol. However, in a recent survey of cyber security ontologies our model is cited as fundamental and suggests that it should serve as a foundation upon which to coalesce existing ontologies and build future ontologies [3]. Furthermore, our model supports the argument that to truly integrate cyber threat intelligence we must escape from representing a complicated and deeply relational activity as a flat and simple list of technical indicators. We argue that to achieve strategic mitigation, intrusion activity must be documented and shared integrating non-technical context while preserving essential and complex relationships.

3 Diamond Model Overview

In its simplest form (Figure 1), the model describes that an *adversary* deploys a *capability* over some *infrastructure* against a *victim*. These activities are called *events* and are the atomic features. Analysts or machines populate the model's vertices as events are discovered and detected. The vertices are linked with edges highlighting the natural relationship between the features. By pivoting across edges and within vertices, analysts expose more information about adversary operations and discover new capabilities, infrastructure, and victims.

An event defines only one step in a series that the adversary must execute to achieve their objective. As such, events are phase-ordered by adversary-victim pair into *activity threads* representing the flow of an adversary's operations. Both *events* AND *activity threads* are

necessary elements of a complete understanding of malicious activity as more effective and strategic mitigation "requires a new understanding of intrusions themselves, not as singular events, but rather as phased progressions." [11]

Once *activity threads* are established, events can then be correlated across threads to identify adversary campaigns, and coalesced into *activity groups* to identify similar events and threats which share common features. These *activity groups* can be used for automated correlation of events as well as for gaming and planning mitigation options and scenarios establishing strategic mitigation plans countering the adversary.

The aforementioned terms and concepts will be further described and discussed in the following sections, beginning with the model's atomic element – the Diamond Event.

4 Diamond Event

Axiom 1 *For every intrusion event there exists an adversary taking a step towards an intended goal by using a capability over infrastructure against a victim to produce a result.*

An **event** defines discrete time-bound activity restricted to a specific *phase* where an *adversary*, requiring external *resources*, uses a *capability* and *methodology* over some *infrastructure* against a *victim* with a given *result*. Of course, not all of the features need to be known to create an event. In almost all cases, most features are expected to be unknown and completed only after the initial discovery as new facts are revealed and more data is gathered.

Core Features The **core features** of an event are: *adversary*, *capability*, *infrastructure*, and *victim*.

Meta-Features The **meta-features** are: *timestamp* (both start and end), *phase*, *result*, *direction*, *methodology*, and *resources*. The meta-features are used to order events within an *activity thread* (§8), group like events in various ways, and capture critical knowledge where possible.

Confidence Value Each event feature, be it a core or meta-feature, has an associated confidence value. This value is left purposefully undefined as each model implementation may understand confidence differently. Furthermore, confidence is likely a function of multiple values, such as the confidence of an analytic conclusion and/or the accuracy of a data source. As necessary, the confidence value may also be itemized as a sub-tuple to better capture the individual elements of confidence.

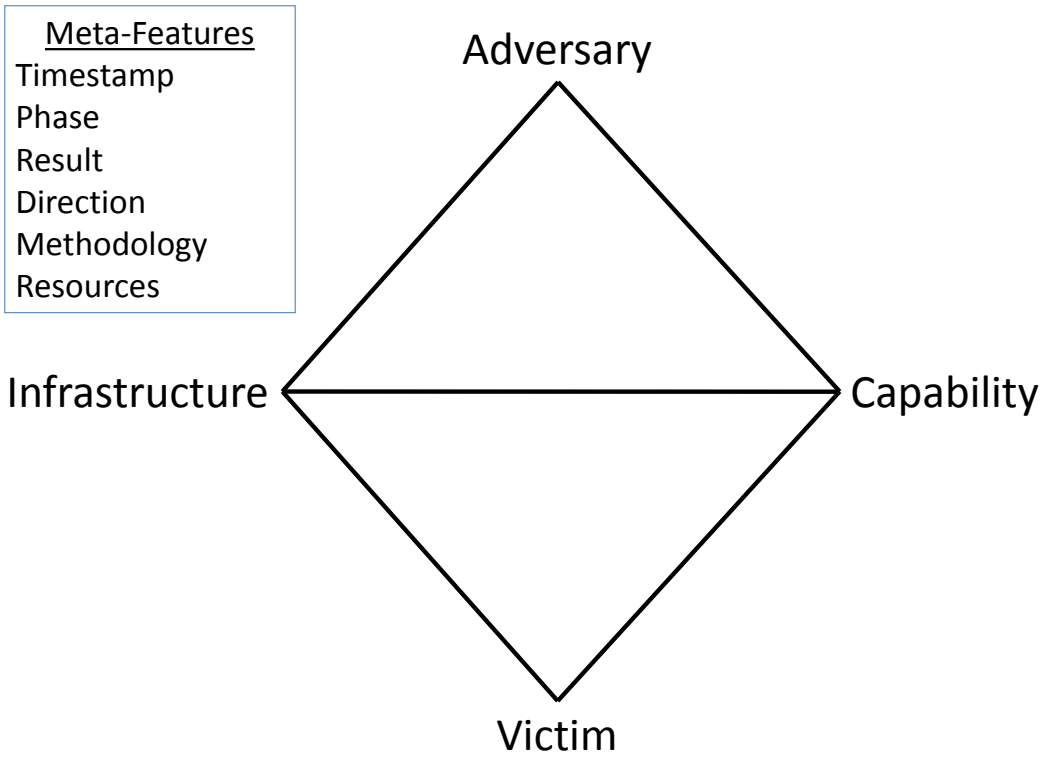

Figure 1: The Diamond Model of intrusion analysis, comprising the core features of an intrusion event: adversary, capability, infrastructure, and victim. The core features are linked via edges to represent the fundamental relationships between the features which can be exploited analytically to further discover and develop knowledge of malicious activity. The meta-features are listed as well, and while not core features, highlights their importance in higher-order analysis, grouping, and planning functions.

One benefit of the model is that it provides an effective (but not necessarily comprehensive) list of features that *should* be present in every event. Therefore, after documenting an event with all available information any empty features are now identified knowledge gaps which should encourage additional pivoting to close those gaps.

An **event**, $\boldsymbol{E}$, is formally defined as a labeled n-tuple where each element of the tuple is knowledge of a feature combined with an independent confidence value.[3]

$$\begin{aligned}\boldsymbol{E} = \langle &\langle Adversary, Confidence_{adversary}\rangle,\\ &\langle Capability, Confidence_{capability}\rangle,\\ &\langle Infrastructure, Confidence_{infrastructure}\rangle,\\ &\langle Victim, Confidence_{victim}\rangle,\\ &\langle Timestamp_{start}, Confidence_{timestamp_{start}}\rangle,\\ &\langle Timestamp_{end}, Confidence_{timestamp_{end}}\rangle,\\ &\langle Phase, Confidence_{phase}\rangle,\\ &\langle Result, Confidence_{result}\rangle,\\ &\langle Direction, Confidence_{direction}\rangle,\\ &\langle Methodology, Confidence_{methodology}\rangle,\\ &\langle Resources, Confidence_{resources}\rangle\rangle\end{aligned}$$

For added flexibility, the basic tuple can be expanded into a hierarchy of nested ordered pairs (referred to as sub-tuples herein for simplicity) to further define a particular feature and capture knowledge for future correlation.

An illustrative example of expanding the victim feature to provide greater definition with information such as: the organization being victimized, the IP address of the host, the name of the host (i.e., hostname), the application which was exploited, and the TCP port which was used to exploit the application:[4]

[3]The event is a variable-sized n-tuple, rather than a fixed size, because the model is not limited to the features defined here and can be expanded to include other elements of interest, like those in the Extended Diamond §5. Once an organization defines their full feature-set the tuple size will be formally defined for that particular instance.

[4]Note that in the example each sub-feature of *Victim* has an independent confidence value, but one could also implement the model where one confidence value is simply applied to all sub-features.

$$\langle Victim, Confidence_{victim} \rangle = \langle \langle Organization, Confidence_{organization} \rangle, \langle HostIPAddress, Confidence_{IP} \rangle, \langle Hostname, Confidence_{Hostname} \rangle, \langle Application, Confidence_{Application} \rangle, \langle TCPPort, Confidence_{TCPPort} \rangle \rangle$$

For analytic purposes, the event can also be understood and represented as a graph as illustrated in Figure 1. In this form the edges represent the natural relationships between the features of an event and identify what is normally visible/discoverable from the perspective of that feature through pivoting (described further in §7). The core features (adversary, capability, infrastructure, and victim) make up a vertex-labeled, undirected, simple graph. A graph-organized event, $\boldsymbol{E}$, is thus defined:

$$\mathbf{E_{vertices}} = \{Adversary, Infrastructure, Capability, Victim\}$$

$$\mathbf{E}_{edges} = \{\{Adversary, Capability\}, \{Adversary, Infrastructure\}, \{Infrastructure, Capability\}, \{Infrastructure, Victim\}, \{Capability, Victim\}\}$$

4.1 Adversary

Axiom 2 *There exists a set of adversaries (insiders, outsiders, individuals, groups, and organizations) which seek to compromise computer systems or networks to further their intent and satisfy their needs.*

An *adversary* is the actor/organization responsible for utilizing a capability against the victim to achieve their intent. Adversary knowledge is generally elusive and this feature is likely to be empty for most events – at least at the time of discovery.

The majority of the time when analyzing the technical aspects of an event, we simply refer to the adversary operator as the adversary. However, the distinction between adversary operator and customer is important to understand intent, attribution, adaptability, and persistence by helping to frame the relationship between an adversary and victim pair. Therefore, we have found these distinctions important:[5]

Adversary Operator This is the actual "hacker" or person(s) conducting the intrusion activity.

Adversary Customer This entity stands to benefit from the activity conducted in the intrusion. It may be the same as the *adversary operator*, or it may be a separate person or group.

For example, a well resourced adversary customer could at different times or simultaneously direct different operators, each with their own capabilities and infrastructure, to a common victim carrying out common or separate goals.[6] To contrast, a lone adversary operator may have access to fewer capabilities and infrastructure points to carry out their activities while also lacking the ability to bypass simple mitigation.

Cognizance of the motivations and resourcing of an adversary operator and their customer, if it exists as a separate entity, will assist in measuring the true threat and risk to the victim resulting in more effective mitigation. Informing these motivations are social-political needs explained later in the Extended Diamond (§5).

4.2 Capability

The *capability* feature describes the tools and/or techniques of the adversary used in the event. The flexibility of the model allows the capability to be described in sufficient fidelity. We intend for capability to be broadly understood and include all means to affect the victim from the most manual "unsophisticated" methods (e.g., manual password guessing) to the most sophisticated automated techniques.

Capability Capacity All of the vulnerabilities and exposures that can be utilized by the individual capability regardless of victim are considered its *capacity*.

[5] While some feature distinctions and categories are suggested throughout this work as useful for everyday analysis, no claim to completeness is made or suggestion that these distinctions form an ontology or taxonomy, nor are these required by the model.

[6] Various activities orchestrated by a higher authority can be modeled and organized within the model as Activity Group Families (§9.7).

Adversary Arsenal An adversary's complete set of capabilities, and therefore the combined capacities of their individual capabilities, is the *adverary's arsenal*.

If the capability capacity is known it should be documented as a sub-tuple of the capability as well as potential paths on an activity-attack graph (§8.3). This initial documentation of an adverary's capabilities can grow over time with Activity Group analysis (§9.5) culminating in the knowledge of their arsenal. This is valuable information in mitigation decisions and planning allowing one to potentially forecast adversary courses of action and reaction.

4.2.1 Command and Control (C2)

Command and control (C2) is the exercise of authority and direction over assets by a commander [33]. In intrusion analysis, this means the channels, communication structures, signals, protocols, and content to or from the adversary intended to cause effect (e.g., gain access, deliberately remove access, exfiltrate data, send attack packets) progressing the adversary towards achieving their goals.

While command and control can take many forms, it is ultimately determined by the capability in use. In terms of analytic pivoting (§7), an analyst pivots over C2 discovering communication between infrastructure and victims. Therefore, for the purposes of our model, command and control is best understood as a sub-feature of capability.

4.3 Infrastructure

The *infrastructure* feature describes the physical and/or logical communication structures the adversary uses to deliver a capability, maintain control of capabilities (e.g., command-and-control/C2), and effect results from the victim (e.g., exfiltrate data). As with the other features, the infrastructure can be as specific or broad as necessary. Examples include: Internet Protocol (IP) addresses, domain names, e-mail addresses, Morse code flashes from a phone's voice-mail light watched from across a street, USB devices found in a parking lot and inserted into a workstation, or the compromising emanations from hardware (e.g., Van Eck Phreaking [34]) being collected by a nearby listening post. We find the following infrastructure role distinctions to be reasonable for most intrusion analysis purposes.

Type 1 Infrastructure Infrastructure which is fully controlled or owned by the adversary or which they may be in physical proximity.

Type 2 Infrastructure Infrastructure which is controlled by an (witting or unwitting) intermediary. Typically, this is the infrastructure that a victim will see as the adversary. It serves to obfuscate the origin and attribution of the activity. Type 2 infrastructure in-

cludes zombie hosts, malware staging servers, malicious domain names, hop-through points, compromised email accounts, etc.

Service Providers Organizations which (wittingly or unwittingly) provide services critical for availability of adversary Type 1 and Type 2 infrastructure (e.g., Internet Service Providers, domain registrars, web-mail providers).

4.4 Victim

A *victim* is the target of the adversary and against whom vulnerabilities and exposures are exploited and capabilities used. As with other features, a victim can be described in whichever way necessary and appropriate: organization, person, target email address, IP address, domain, etc. However, it is useful to define the victim persona and their assets separately as they serve different analytic functions. Victim personae are useful in non-technical analysis such as cyber-victimology (§5.1.2) and social-political centered approaches (§5.1) whereas victim assets are associated with common technical approaches such as vulnerability analysis.

Victim Persona *Victim Personae* are the people and organizations being targeted whose assets are being exploited and attacked. These include organization names, people's names, industries, job roles, interests, etc.

Victim Asset *Victim Assets* are the attack surface and consist of the set of networks, systems, hosts, email addresses, IP addresses, social networking accounts, etc. against which the adversary directs their capabilities. Victim assets often exist both inside and outside a persona's control and visibility but are still available for targeting by an adversary. Common examples of this include webmail accounts and cloud-based data storage.

A victim asset can be the end target (e.g., victim) in one event and then leveraged as infrastructure in further events (likely Type 2 Infrastructure as described previously in §4.3). In this way, one must always beware that the apparent target of activity may not necessarily be the victim.

4.4.1 Vulnerabilities and Exposures

Axiom 3 *Every system, and by extension every victim asset, has vulnerabilities and exposures.*

Adversary capabilities exploit the vulnerabilities and exposures defined by Axiom 3 to fulfill their intent. The model's flexibility allows these to be defined as a sub-feature of the victim. These can be described as broadly as "lack of user education causing email-borne hyperlinks to be clicked" or as specific as a CVE [35] to fit the documentation requirements of the event.

Victim Susceptibilities The set of vulnerabilities and exposures of a victim susceptible to exploitation is referred to as the *victim susceptibilities*.

In our model the list of victim susceptibilities are easily expressed as a sub-tuple of the victim. This information is valuable when compared to capability capacity and adversary arsenal (§4.2) to determine mitigation options. As with capability capacity, this can be alternatively or conjunctively described using activity-attack graphs (see §8.3).

4.5 Event Meta-Features

The event meta-features expand the model slightly to include non-critical, but important, elements of Diamond events. The meta-features described here are those which we find most useful, but the model is not limited to these. Those who implement or extend our model may wish to add additional meta-features to capture other critical elements of information associated with an event.

4.5.1 Timestamp

Each event is notated with a date and/or time that it occurred. It can be as specific as necessary or expressed as a range of values indicating the start and stop time of the event. Timestamps are an integral part of grouping malicious activity as the timestamp allows for a reduced confidence in knowledge over time (i.e., a decay function) as the likelihood of adversary changes increase over time. Furthermore, timestamps combined with a collection of adversary events over time can lead to other unique forms of analysis, such as establishing periodicity and pattern-of-life deduction as in [6].

4.5.2 Phase

Axiom 4 *Every malicious activity contains two or more phases which must be successfully executed in succession to achieve the desired result.*

Malicious activity does not happen in a single event but rather two or more events. Others have provided sufficient evidence to support the conclusion that all intrusion activity should be considered as a chain of events [11, 36, 15]. For example, first an adversary must find the

victim (usually using research and/or scanning) then discover a vulnerable host, followed by exploitation, the establishment of command-and-control, and lastly by operations of some sort. Often, different capabilities and sometimes different infrastructure are used through the phases of an intrusion. At a minimum a victim must first be identified (which could be as simple as the selection of a random IP address) and then an action performed.

Our model can utilize any phased model of adversary operations (such as [11, 36, 15]).[7] Later, this becomes a significant feature as the *phase* of the event describes its location in the activity thread (§8).

While Axiom 4 guarantees a set of phases for every activity, there has been no consensus or evidence that there exists a set of phases satisfying the characterization of all malicious activity. In fact, the existence of so many multi-phased activity definitions suggests otherwise, for example [36, 11]. Therefore, we will assume that users of the Diamond may define non-essential phases for some activity.

Formally, the phases, $\boldsymbol{P}$, are defined as an ordered n-tuple, where n is the number of phases a model user has defined as necessary and sufficient to describe all possible events and the phase of each event is one, and only one, element of the ordered tuple $\boldsymbol{P}$:

$$P = \langle p_1, p_2, \ldots, p_n \rangle$$

Where:

- $n \geq 2$ (there exists at least two phases as required by Axiom 4)
- p is a phase in the chain of adversary operations
- p_1 is the first phase of an adversary's operations
- p_{n+1} is a phase executed subsequent to p_n

4.5.3 Result

While the results and post-conditions of an adversary's operations will not always be known, or have a high confidence value when they are known, they are useful to capture. It is particularly useful to look across an adversary's operations to determine their success rate with particular capabilities or against sub-sets of victims. A collection of post-conditions can also provide a broader view of adversary intent. There are several ways of potentially documenting the result. One method is to use the 3-tuple $\langle Success, Failure, Unknown \rangle$. Another is to separate it by security fundamentals: Confidentiality Compromised, Integrity

[7] The identification of a phase for each event is not essential to maintain knowledge and correlate events, but is useful for mitigation planning purposes and Kill Chain analysis.

Compromised, and Availability Compromised. While another approach could document all of the post-conditions resulting from the event, such as targeting information gained (in the reconnaissance stage) or passwords exfiltrated later useful in masquerade attacks. Furthermore, one could use an existing taxonomy for attack results such as the categories Cohen describes in [26].

4.5.4 Direction

The *directionality* of an event is important when mitigation options and the placement of detection is considered. This meta-feature is typically useful when describing network-based events, but can also be useful to describe host-based events as well. There are generally seven potential values for this feature: Victim-to-Infrastructure, Infrastructure-to-Victim, Infrastructure-to-Infrastructure, Adversary-to-Infrastructure, Infrastructure-to-Adversary, Bidirectional, or Unknown. By maintaining this information and considering the adversary's activity direction over time better decisions on which combination of external-only, external-facing, or internal-facing detection and mitigation actions would work best to counter the adversary.

4.5.5 Methodology

The *methodology* meta-feature allows an analyst to describe the general class of activity, for example: spear-phish email, content-delivery attack, syn flood, port scan, etc. As with other feature types, this also allows more than one definition as necessary. For instance, a malicious spear-phishing email with malware attached may be categorized as both a "spear-phish email" and a "content-delivery attack." Whereas a spear-phishing email with a hyperlink leading the user to a malicious website may be categorized as both a "spear-phish email" and "user-redirect exploit." This method better categorizes events and allows indicator-independent event comparison both for a single adversary and across adversaries for grouping (§9) and mitigation purposes.

Several existing taxonomies could easily be incorporated into this feature reducing effort and increasing interoperability with existing frameworks. Some examples include Snort classtypes [37] and many more formal studies [25, 29, 26, 28].

4.5.6 Resources

Axiom 5 *Every intrusion event requires one or more external resources to be satisfied prior to success.*

The *resources* meta-feature lists one or more external resources the event requires to be satisfied. Resources are to be broadly understood as any and all supporting elements on which the event, and therefore each core- and meta-feature, depends. This meta-feature becomes important when resource-constraint and center-of-gravity mitigation strategies are considered as well as the identification of knowledge gaps and hypothesis testing as described later in §8.2.

Obviously, this meta-feature could be envisioned as encompassing an intractable number of elements. However, as with the other features the Diamond Model does not require completeness, only sufficiency. Therefore, an organization only needs to enumerate the resources necessary for their implementation to be effective for their particular use(s).

Example resources include:

- Software (e.g., metasploit, operating systems, virtualization software)
- Knowledge (e.g., how to run metasploit, where to obtain exploits)
- Information (e.g., a username/password to masquerade)
- Hardware (e.g., workstations, servers, modems)
- Funds (e.g., credit to purchase domains)
- Facilities (e.g., electricity, shelter)
- Access (e.g., a network path from the origin host to the victim and vice versa, a routable IP address and network access from an Internet Service Provider (ISP))

4.5.7 Meta-Feature Expansions

Several meta-features have been described which work well integrated within the model. There are many other meta-features to a malicious intrusion event which can be considered for inclusion based on the needs of the organization: *data source* (the source of the data which captured or detected the event), *author* (the analyst-author of the event), *detection method* (the tool, technique or capability which detected the malicious event), *detection signature* (the signature or heuristic which detected the malicious event), etc. Adding additional meta-features will enhance the model by allowing users, analysts, and organizations to maintain important information associated with an event for future use (such as effective sourcing or credit for discovery/authorship, refining analytics, understanding confidence intervals, quality control, etc.).

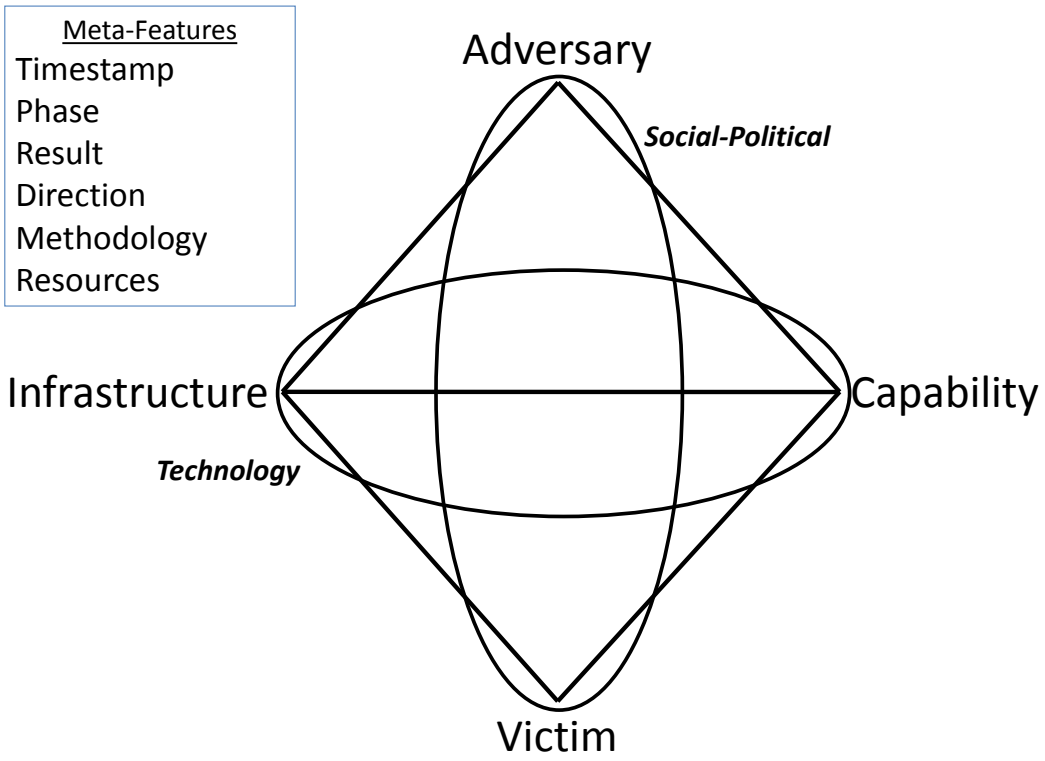

Figure 2: The extended Diamond Model illustrates the unique Social-Political and Technology features. These features highlight special relationships between the Adversary-Victim (through the social-political needs, aspirations, and motivations of the adversary and the ability of the victim to fulfill those needs) and the Capability-Infrastructure (through the technology used to enable their communication).

5 Extended Diamond Model

As described earlier, the Diamond Model easily extends to include other necessary features. As illustrated in Figure 2, two additional fundamental meta-features of any intrusion activity are: the *Social-Political* meta-feature determining the Adversary-Victim relationship, and the *Technology* meta-feature enabling both the infrastructure and capabilities. These two unique features overlay two other features inextricably defining a relationship: one laid across the adversary-victim axis, the other across the capability-infrastructure axis.

5.1 Social-Political

Axiom 6 *A relationship **always** exists between the Adversary and their Victim(s) even if distant, fleeting, or indirect.*

Adversary-Victim pairs are predicated on a producer-consumer relationship which are underpinned by the social-political needs and aspirations of the adversary (e.g., to generate income, to gain acceptance in the hacker community, to become a hegemon, to increase business profits).[8] The relationship denotes the need(s) of the adversary and the ability of the victim to satisfy the need(s) defining adversary intent (e.g., economic espionage, traditional espionage, criminal fraud, denial of service attack, website defacement). The victim unwittingly provides a "product" (e.g., computing resources & bandwidth as a zombie in a botnet, a target for publicity, industrial or business sensitive information for economic espionage, financial information and username/passwords for fraud) while the adversary "consumes" their product.

Intent Although *intent* is a critical aspect of understanding intrusion activity and should strongly inform mitigation decisions, it is not included as a fundamental top-level meta-feature of the Diamond, but fits better as a feature in a social-political sub-tuple further allowing one to hypothesize higher-order needs and aspirations.

5.1.1 Persistent Adversary Relationships

Axioms 2 and 6 can be combined to state that there are adversaries and they inexplicably establish a relationship with their victim(s). However, not all Adversary-Victim relationships are equal. Some adversaries execute "smash and grab" operations without concern for access or data beyond what is immediately available unconcerned about losing access at anytime. Other adversaries doggedly persist in their efforts against some victims even in the face of substantial mitigative action. Some adversaries even go so far as to retaliate against those who mitigate their activities [6]. The persistence of the adversary to continue to obtain access and/or information from a victim is one possible, albeit important, characterization of the Adversary-Victim relationship.

As such, and given evidence of both persistent and non-persistent relationships,the following Axiom can be posited:

[8]The term social-political is carefully chosen to categorize the broad set of needs and aspirations by the majority of adversaries which includes, but is not limited to, individuals, partnerships, loosely-organized collectives, hierarchical groups, non-state, and state actors. Their needs can be broadly described in both social and political terms. One could argue that politics is an extension of social needs, that of individuals organizing under authority to satisfy collective desires. However, we feel that separating these two terms produces the best balance of dialog on individual/non-state group needs with the needs of authority (e.g., government or military) and how those needs inform victim selection and therefore mitigation decisions.

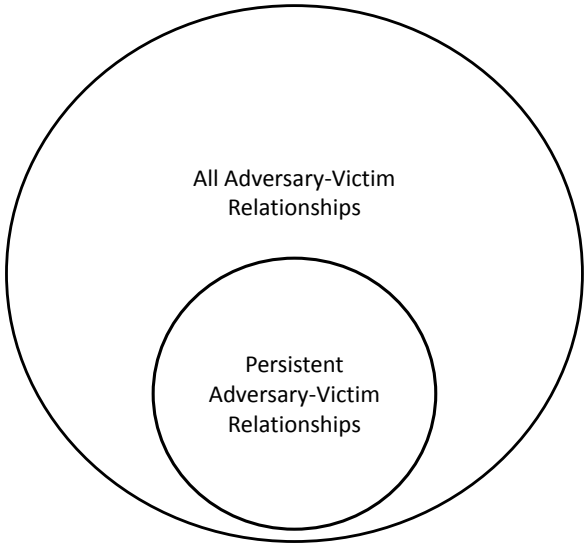

Figure 3: A Venn diagram illustrating the set of all adversary-victim relationships as defined by Axioms 2 and 6 as well as the sub-set of persistent adversary relationships defined by Axiom 7.

Axiom 7 *There exists a sub-set of the set of adversaries which have the motivation, resources, and capabilities to sustain malicious effects for a significant length of time against one or more victims while resisting mitigation efforts. Adversary-Victim relationships in this sub-set are called* ***persistent adversary relationships****.*

Persistent Adversary A *persistent adversary* is an adversary that satisfies Axiom 7 in a particular Adversary-Victim relationship.

Figure 3 illustrates the relationship between the complete set of Adversary-Victim relationships defined by Axioms 2 and 6 and the sub-set of persistent adversaries as defined by Axiom 7. The placement of an Adversary-Victim relationship into one of these sets is determined by the satisfaction of Axiom 7.

It is not necessary that because an adversary is persistent against one victim they are persistent against all victims. For instance, in one activity the adversary may gain access, determine there is no value, and leave without any regard to persistence. However, in another activity the adversary may persist for much longer in order to gain more value. From the other perspective, a victim may be host to multiple adversaries of which some may be persistent while others are non-persistent. Therefore, the persistence or non-persistence is determined by the particular Adversary-Victim pair.

Figure 4: Degree of Persistence Spectrum illustrates that not all adversary persistent relationships are equal but instead fall on a spectrum between fleeting and enduring. Where a particular adversary-victim relationship falls on the spectrum is a function of many elements and also changes over time.

Furthermore, persistence is not a binary nor static characteristic. While it is well-known that many persistent intrusions can be mitigated by technical measures, such as in Stoll [4], Cheswick in "Berferd" [6] illustrates that some adversaries resist technical measures and even public shaming attempts. In the case of "Berferd", mitigation was ultimately achieved by a phone call to the hackers' mothers. Therefore, the degree of persistence varies and we propose the following corollary:

Corollary 1 *There exists varying degrees of adversary persistence predicated on the fundamentals of the Adversary-Victim relationship.*

The *degree of persistence* describes the strength of the adversary's motivation and capabilities as well as the effort and resources an adversary will expend to maintain their effect. The degree of persistence manifests itself along a spectrum between fleeting to the most enduring, as illustrated in Figure 4, and in many cases determines the amount of effort and resources a defender requires to resist persistence. The stronger the motivation and capability of the adversary and the more resistant they are to mitigation equates to a more enduring persistent relationship moving to the right of the spectrum.

Traditionally, mitigation has been limited to technical means revolving around the capability of the adversary and has had little impact on their motivation and resources resulting in the adversary returning shortly after being removed from the victim. By making the social-political relationship and its associated needs and aspirations a key part of malicious activity, the Diamond enables the application of non-traditional domains such as psychology, criminology, victimology, marketing, consumer behavior, and economics to expand mitigation options. Particularly, it underpins the decision making of the adversary and their perceived preference highlighting variables and aspects which can be controlled and influenced in favor of the defender in addition to traditional technical options.

The following are some of the elements of the Adversary-Victim relationship which determine the degree of persistence:

- The relative strength of the adversary's needs which the victim fulfills compared to other needs
- The risk the adversary perceives in continued effects

- The adversary's cost required to maintain effect
- The uniqueness of the victim to satisfy a particular need
- The continued satisfaction of the need by the victim
- The level of effort and resources a defender expends to resist persistence

For both persistent or non-persistent adversary relationships, placement on the spectrum is unique to each Adversary-Victim pair. For ease of reference and analysis, and without disregard to the complexity and continuum the spectrum represents, we generally consider two classes of victim on the spectrum: *victims of opportunity* and *victims of interest*.

Victim of Opportunity A victim who is an expendable commodity in an adversary's operations where loss of access would likely not be noticed nor cause the adversary to spend resources regaining access. Victims in this class fall to the left-side of the persistence spectrum towards "fleeting" as well as within the non-persistent relationship set. These victims were likely initially targeted because they were vulnerable and available at the right time.

Victim of Interest A non-expendable commodity where continued access provides value enough to an adversary that the loss of access would cause notice and the adversary would spend resources regaining access to that or related victims. Victims in this class fall to the right-side of the persistent spectrum towards "enduring."

Importantly, a persistent Adversary-Victim relationship is not static on the spectrum – it can shift. Just because a victim initially starts as fleeting and a victim of opportunity does not mean that they cannot shift later. For example, if a victim is initially exploited with a self-propagating worm but the adversary finds the victim is of higher value than just a commodity, they could become a victim of interest moving right along the spectrum towards "enduring."

5.1.2 Cyber-Victimology

Our model is unique in that it places the victim and adversary in an equivalent space and highlights the usually unspoken relationship between the two. Furthermore, as our model expands to encompass many adversaries and victims through *Activity Threads* (§8) and *Activity Groups* (§9) we can begin to draw in expertise from criminology and victimology leading to important questions such as:

- Why was a particular entity victimized?

- Is there a common set of victims?
- Do the victims share a common trait?
- Can we deduce intent from the set of victims?
- Who might be other, yet unknown, victims?
- Who has the needs and intent to victimize this set of organizations?

Importantly, with a better victimology model, we can begin to examine methods of countering the adversary by making victims less appealing and predicting future victims. This allows an organization to maximize detection resources appropriately, just as a detective focuses on the highest risk population and area of concentrated crime rather than patrolling random areas.[9]

Recent "watering-hole" attacks[10] illustrate how adversaries use this concept to profile their victims in order to place an exploit in the most lucrative place. For example in April 2013 a recent exploit staged on Tibetan activist-related websites attempted to exploit any visitor with a vulnerable browser [40]. However, alternatively, if the Social-Political feature is used effectively in conjunction with the Victim-Centered approach (§7.1.1), some watering-hole locations can be predicted and targeted detection/mitigation put into place to preempt the malicious activity.

5.1.3 Shared Threat Space

If two or more victims share enough features which would satisfy the needs of one or more adversaries then they are in a "shared threat space." Early identification of shared threat space is a cornerstone for strategic and proactive mitigation. For example, targeted attacks against one member enable the collective threat space to forecast and predict future attacks. Furthermore, sharing of threat intelligence is more lucrative with those most likely to be impacted by a similar adversary.

5.2 Technology

In addition to the social-political meta-feature, the *technology* meta-feature also highlights a special relationship and spans two core features: capability and infrastructure. This

[9] An interesting criminology study revealed that an increase in tree cover within urban neighborhoods statistically correlated to a reduction of crime [38]. Could a potential parallel exist for intrusion activity?

[10] Watering-hole attacks are a methodology where the adversary compromises legitimate websites which they believe their intended class of victims will visit thereby exploiting them. The analogy and term is drawn from lions laying in wait to ambush prey at a watering hole [39].

represents the technology connecting and enabling the infrastructure and the capability to operate and communicate.

For example, if installed malware resolves domains and communicates over HTTP, the technologies used are: Internet Protocol (IP), Transport Control Protocol (TCP), Hypertext Transport Protocol (HTTP), and the Domain Name System (DNS). By analyzing technology and its potential anomalies/misuse, an analyst discovers new malicious activity regardless of the underlying infrastructure and capability (also known as the *technology-centered approach* §7.1.6). Furthermore, understanding the technologies involved in adversary activity assist in the identification of the most appropriate detection locations, data types, and capabilities.

6 Contextual Indicators

Indicators are those elements of information used by systems and analysts to detect adversary operations. In the normal course of business, indicators are loaded into detection systems which alert analysts to potential adversary activity. Traditional indicators have been limited to technical details. Some have extended these to include additional metadata [32]. However, it is time that indicators extend to include elements that are non-technical, behavioral, and conceptual in nature which augment, but are not directly implemented by, automated detection.

Contextual Indicator A *contextual indicator* is an element of information placed into the context of an adversary's operations enriching both detection *and* analysis. Diamond-derived contextual indicators ensure the relationship between elements and their role are retained and analytic concepts such as adversary needs and intent are fully incorporated producing a more complete context.

For instance, in a traditional indicator approach an adversary's infrastructure IP address is a common element. Using our model as a basis for an ontology, this IP address can be placed into context providing the analyst not only knowledge of adversary infrastructure (likely a detection alert) but also knowledge of types/classes of victims previously compromised and possibly which items of information the adversary was attempting to compromise (e.g., business planning documents, intellectual property). Using this enhanced knowledge, the analyst is now armed to not only detect and confirm the intrusion (available with traditional indicators), but also determine if they are part of an adversary campaign, information likely to be targeted by the adversary, and the adversary's intent and social-political needs potentially forecasting future adversary operations (§5.1).

This context enables the organization to take much more strategic mitigation. For example, the organization can now enable adversary-specific detection and mitigation on assets which contain information of value, develop a protracted mitigation campaign (such as one described in [11]), identify and communicate with partners in the *shared threat space* (§5.1.3)

to develop joint mitigation plans, and share non-technical indicators, etc.

7 Analytic Pivoting

Pivoting is the analytic technique of extracting a data element and exploiting that element, in conjunction with data sources, to discover other related elements. Ultimately, pivoting is about the fundamental analytic task of hypothesis testing. Each element of an intrusion event generates its own hypotheses which require evidence to strengthen, weaken, or change the hypothesis. Pivoting is the task of discovering related elements (evidence) which inform the hypothesis and also generate new hypotheses themselves. Pivoting success relies on the analyst to understand the relationship between elements and their ability to successfully exploit a data element and data sources (e.g., if I have this information combined with this data source then I can find this...).

The Diamond Model fundamentally supports analytic pivoting and is one of its strongest characteristics. In fact, the Diamond was originally revealed after exploring pivot scenarios. The core features are structured as a 'diamond' with connecting edges highlighting pivot opportunities to illuminate other elements of an adversary's operations. With one point of the Diamond the analyst can possibly discover and develop the other connected features.[11]

Using Figure 5 as an example: (pivot 1) a victim discovers malware on their network, (pivot 2) the malware is reversed exposing the command-and-control (C2) domain, (pivot 3) the domain is resolved exposing the underlying IP address hosting the malware's controller, (pivot 4) firewall logs are examined illuminating other compromised hosts in the victim's network establishing communications with the now-revealed malware controller IP address, and finally (pivot 5) the IP address registration reveals adversary details providing potential attribution of the adversary.

7.1 'Centered' Approaches

The model lends itself to several focused intrusion analysis tradecraft concepts. These are referred to as 'centered' approaches as they are centered on a specific feature of the Diamond in order to discover new malicious activity and reveal activity related to the other connected features and the feature itself.

7.1.1 Victim-Centered Approach

Most organizations, through normal network and host monitoring, detection, and defense operations, are exercising a victim-centered approach. With this approach, analyzing data

[11] Success is not guaranteed by the Diamond. It only highlights what is possible, not what is certain.

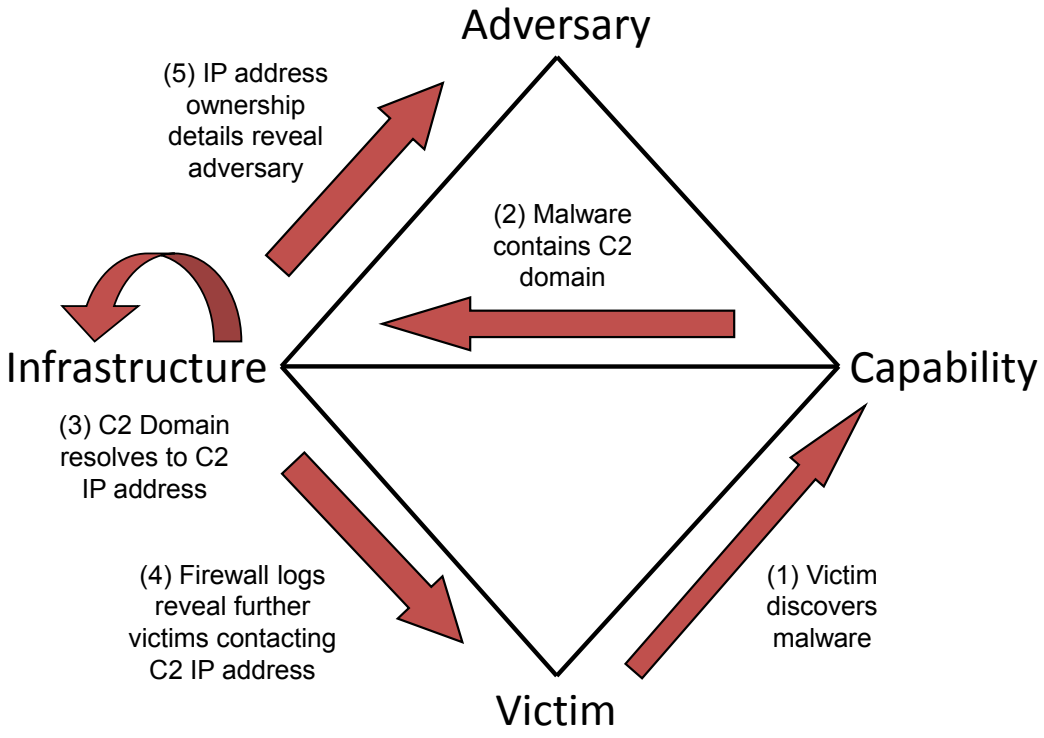

Figure 5: Analytic pivoting using the Diamond is illustrated. One of the most powerful features of the Diamond, pivoting allows an analyst to exploit the fundamental relationship between features (highlighted by edges between the features) to discover new knowledge of malicious activity.

related to a potential victim reveals the other related (and Diamond-connected) elements: malicious capabilities and infrastructure. The Honeynet Project is an excellent example of this approach. By establishing a specially configured host intended to be victimized they are inviting adversaries to exploit the host, revealing their capabilities and infrastructure which can then be publicized for mitigation and education [7].

Another interesting example of the victim-centered approach is where analysts monitored services for Himalayan users thought to be targeted by a highly capable adversary [41]. This, as predicted by the Diamond Model, produced new information about malicious capabilities and infrastructure as the adversary attacked the users of the monitored network. Interestingly, this victim-centered approach was combined with the social-political-centered approach (§7.1.5) allowing the researchers to target a specific adversary by predicting their victim, increasing their chances of success and adding attribution confidence.

7.1.2 Capability-Centered Approach

The capability-centered approach exploits features of a capability to discover those other elements related in adversary operations: victims whom that capability is used against, infrastructure supporting the capability, technology enabling the capability, clues to other related capabilities, and (possible) clues to the adversary. The results of this approach are most commonly seen with anti-virus vendor reports.

As a first example, analysis by Symantec and CrySyS provided a link from Stuxnet to Duqu based on several common features and techniques employed in the code suggesting a common author. In this case, those features were so advanced it led them to pivot up to the adversary feature to deduce potential adversaries responsible. This is an example of a capability to adversary pivot, using the social-political meta-feature to strengthen confidence in attribution [42].

As a second example, Kaspersky's analysis of "Red October" provides an excellent case-study in capability-centered analysis with multiple pivots. Here the work begins with the malware capability and is reverse engineered for technology (HTTP, RC4 encryption, zlib compression), C2 structures, and infrastructure. The capability was then used in combination with their anti-virus detection database from victims (victim-to-capability pivot) to detect "over 1000 different" associated files which were also reversed to identify other infrastructure (capability-to-infrastructure pivot) which were then "sinkholed"[12] to identify global victims (infrastructure-to-victim pivot). Each victim was then further identified as to their social-political position (e.g., embassy, government, military, energy) presumably to allow the reader to infer potential adversaries which would have matching social-political needs using cyber-victimology (§5.1.2) [43].

[12] "Sinkholing" is an aggressive defender technique to takeover positions of the adversary infrastructure for mitigation (the adversary can no longer use what it does not control) and analysis (malware and victims continue to communicate to the now defender-controlled infrastructure).

7.1.3 Infrastructure-Centered Approach

The infrastructure-centered approach focuses on the malicious infrastructure of the adversary. From this element other related elements can be discovered: victims in contact with the infrastructure, capabilities being delivered or controlled with the infrastructure, other related infrastructure (such as IP addresses resolved by malicious domains), and (possible) clues to the adversary including those who may be in direct control of the infrastructure.[13]

The Command Five team demonstrated a heavily infrastructure-centered approach in their SKHack investigation [44]. While the initial details were gleaned from malware discovered during the response, the authors used the resolutions of known callback domains to IP addresses and then pivoted to the WHOIS registration information to discover many other domains with a common registrant (infrastructure-to-adversary pivot). They then successfully mapped infrastructure which had not been used in the attack but was likely controlled by the same adversary positioning for preemptive defensive actions (e.g., blocking network access to those domains prior to their operational use). Further research on the registered domains also yielded information on malware used in other attacks against different victims but was also likely used by the same adversary (infrastructure-to-capability pivot).

7.1.4 Adversary-Centered Approach

One could theorize that the adversary-centered approach is the most difficult of the various centered-approaches. It involves monitoring an adversary directly to discover their infrastructure and capabilities. Of course, this will likely be the most fruitful approach but is limited by the need for access. For instance, the US Federal Bureau of Investigation (FBI) monitored the phone-calls and modem activity of the "Phonemasters" hacking group identifying the full scope of their operations including other adversary personae involved as well as their victims, capabilities, and infrastructure [45]. However, one must be cautioned by the tales of others who track adversaries too closely and pay a price [46].

7.1.5 Social-Political-Centered Approach

The social-political-centered approach is unique. Alone, it does not lead directly to new elements or indicators but rather capitalizes on an expected adversary-victim relationship to hypothesize who may be a victim and what may be their adversaries, or alternatively who may be an adversary and their expected victims. This can then lead to elements which can be exploited using the adversary-centered or victim-centered approach to gain tactical details.

[13] Analysts often exploit the adversary-infrastructure link by exporting registration information but are often foiled by false information. However, false information (e.g., such as that in a domain registration) can be useful if the adversary uses the information consistently providing a common persona that can be tracked and/or traced between malicious events

Analytic conclusions drawn from correlating intrusion activity and real-world political events are actually quite common. As early as 1990 Cheswick correlated intrusion activity against his network to the 1990-91 Gulf War [6]. More recently the 2008 Georgia DDoS attacks and sustained attacks against Pro-Tibet groups have been correlated to current political events [47, 48]. However, the everlasting caution that correlation is not causation must be heeded.

7.1.6 Technology-Centered Approach

The technology-centered approach allows an analyst to target potential misuse or anomalous use of a technology to discover previously unknown infrastructure and capabilities which utilize such techniques. Monitoring and detecting anomalies in the Domain Name System (DNS) has been a popular and fruitful method of implementing the technology-centered approach to discover new malicious activity [49, 50]. Others have explored anomalies in packet headers on back-bone networks [51].

8 Activity Thread

Axiom 4 states that an adversary does not operate in a single event against a victim, but rather in a chain of causal events within a set of ordered phases in which, generally, each phase must be executed successfully to achieve their intent.[14] An *activity thread* is a directed phase-ordered graph where each vertex is an event and the arcs (i.e., directed edges) identify causal relationships between the events. The arcs are labeled with the analytic confidence establishing the causal relationship, whether the path is AND (necessary) or OR (optional – there is more than one potential path from an event), whether the arc is actual or hypothesized, as well as with the information or resource the preceding event provides that is required for the next event to occur.[15] The threads are organized vertically such that each thread describes all of the causal events an adversary executed against a specific victim (however the implementation of the model defines the victim feature) collectively aimed at fulfilling the adversary's intent. Therefore, each thread is specific to one adversary-victim pair – although in many cases activity threads may only vary slightly amongst victims as

[14] As stated in §4.5.2, the set of phases can included non-essential phases for a given activity and therefore not all activity may conform to the complete set of phases available. Therefore, we say that *generally* each phase must be successfully executed to achieve an intent, but it does not necessarily hold for all phases across all activity.

[15] Both AND/OR and requires/provides concepts have been incorporated from previous models and both are useful in different modes of mitigation strategy development. Conjunctive and disjunctive attack paths are borrowed from Schneier's original work on Attack Trees [15] and is useful for reachability, path optimization, and other graph analysis techniques to develop mitigation strategies. The concept of resource-focused attack graphs is borrowed from [52] and useful in resource constraint mitigation strategy development. This is not to say that both must be used together but rather provides maximum opportunity to apply different techniques for further comparison as neither has been shown to be optimal by themselves in generating mitigation strategies. The outputs of these techniques can then be compared using a decision support model such as ADAM to weigh their various risks, costs, and benefits [13]. To support this, and as described originally by Schneier in [15], the arcs can also include weighting, priority, or other quantifiers.

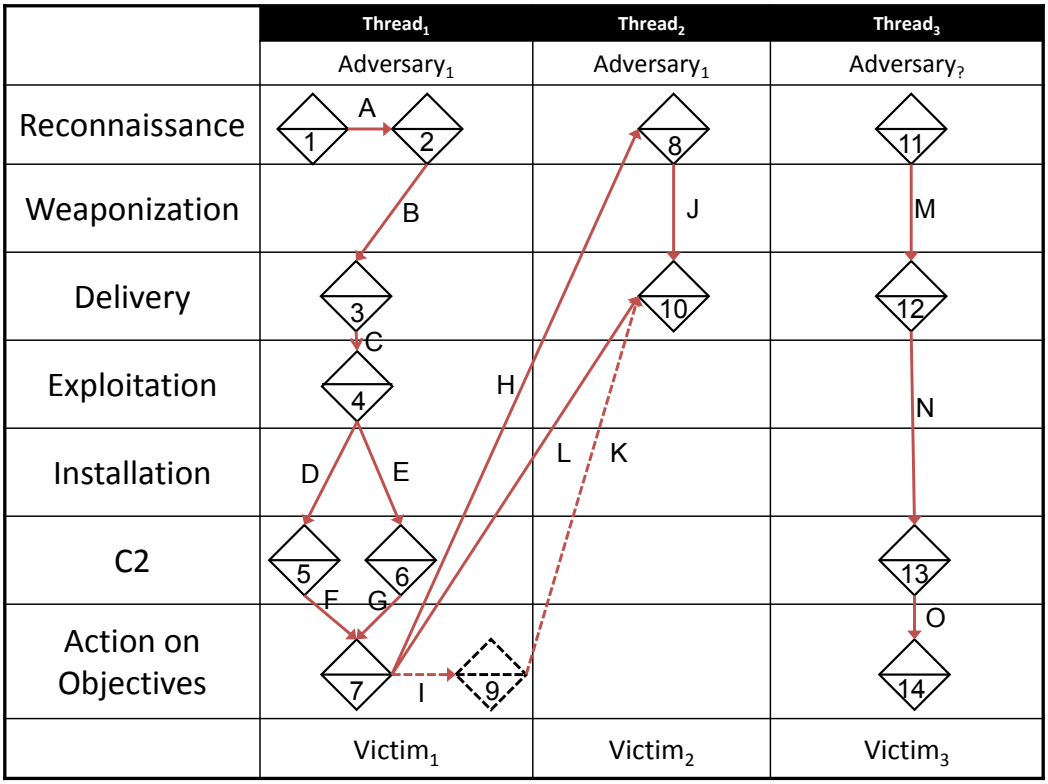

Figure 6: An example visualization of activity threads illustrating Diamond events being linked vertically (within a single victim) and horizontally (across victims) via directed arcs designating a causal relationship between the events (i.e., this event occurred because of, and subsequent to, this event). In the figure, solid lines represent actual element of information supported by evidence and dashed lines represent hypothesized elements. See Table 1 for Event Descriptions and Table 2 for Arc Descriptions.

Table 1: Example Activity Thread Event Descriptions for Figure 6

Event	Hypothesis/Actual	Description
1	Actual	Adversary conducts a web search for victim company Gadgets Inc. receiving as part of the results their domain gadgets.com
2	Actual	Adversary uses the newly discovered domain gadets.com for a new search "network administrator gadget.com" discovering postings to forums from users claiming to be network administrators of gadget.com revealing their email addresses
3	Actual	Adversary sends spear-phish emails with a trojanized attachment to the network administrators of gadget.com revealed in Event 2
4	Actual	One network administrator (NA1) of gadet.com opens the malicious attachment executing the enclosed exploit allowing for further code execution
5	Actual	NA1's host exploited in Event 4 sends an HTTP Post message to an IP Address registering it with a controller and receives an HTTP Response in return
6	Actual	It is revealed from reverse engineering the malware on NA1's host that the malware has an additional IP address configured which acts as a back-up if the first host does not respond
7	Actual	Through a command-and-control HTTP response message sent to NA1's host, the malware begins to proxy TCP connections
8	Actual	Through the proxy established on NA1's host, Adversary1 does a web search for "most important research ever" and finds the victim Interesting Research Inc.
9	Hypothesis	Adversary1 checks NA1s email contact list for any contacts from Interesting Research Inc. and discovers the contact for the Interesting Research Inc. Chief Research Officer
10	Actual	Chief Research Officer of Interesting Research Inc. receives an spear-phish email from Gadget Inc's NA1's email address sent from NA1's host with the same payload as observed in Event 3
11	Actual	An unknown adversary scans for vulnerable web servers including Victim3
12	Actual	An exploit for a vulnerability scanned for previously in Event 10 is delivered to Victim3 via the network
13	Actual	The exploited server, Victim3, establishes a remote shell to the adversary
14	Actual	The adversary uses the remote shell to download all of the documents in Victim3's private directory

Table 2: Example Activity Thread Arc Descriptions for Figure 6

Arc	Confidence	And/Or	Hypothesis/Actual	Provides
A	Low	And	Actual	Provides the domain for Gadgets Inc., gadgets.com
B	High	And	Actual	Provides spear-phishing targets: email addresses for network administrators of gadgets.com
C	High	And	Actual	[None]
D	High	Or	Actual	[None]
E	High	Or	Actual	[None]
F	High	And	Actual	[None]
G	High	And	Actual	[None]
H	Medium	And	Actual	Provides proxy access from previous victim to search engine
I	Low	And	Hypothesis	Access to email contact list
J	High	And	Actual	Victim organization identification
K	Low	And	Hypothesis	Victim email address, name, and role identification
L	High	And	Actual	Spear-phish trojanized email
M	High	And	Actual	Provides the output of the successful scan results identifying the victim web server as vulnerable to the exploit
N	High	And	Actual	[None]
O	High	And	Actual	Provides the established remote shell

the adversary consolidates infrastructure, processes, and capabilities to reduce cost.

Vertical Correlation It is rarely the case that all events in a single vertical activity thread are known. Furthermore, it may take effort to establish causal relationships between events within a thread requiring additional research, data gathering, and analysis. The analytic process of identifying knowledge gaps, filling those gaps with new knowledge, and establishing causal relationships (and associated arc labels) within a single vertical adversary-victim activity thread is referred to as *vertical correlation*. By phase-organizing the thread, one can also more easily identify knowledge gaps where activity should have occurred but no knowledge of such exists (see §8.2 for more information).

It is common for an adversary to use resources gained in one operation to enable future operations or to exploit internal trust relationships to gain deeper access into a specific network – known in penetration testing as pivoting and lateral exploitation. Therefore, causal relationships (arcs) can span one or more threads horizontally. Furthermore, as seen in Figure 6, phases can contain more than one event and arcs can even go 'backwards' to describe an iterative process while the edges describe the resources obtained and used between events.

Horizontal Correlation The analytic process of causally linking events between vertical threads across adversary-victim pairs, identifying common knowledge gaps between threads, and using knowledge from one thread to fill knowledge gaps in another is referred to as *horizontal correlation*. This process also leads to the identification of common features across victims which can lead to the creation of an activity group in a process defined later (§9).

These activity threads form a new type of phase-ordered attack graph[16] informed by observations of actual events to predict likelihood and adversary preference for particular paths. As with traditional attack graphs, activity threads model complex multi-stage activity which may exploit multiple system and network vulnerabilities. However, unlike traditional attack graphs which attempt to exhaustively list all possible paths, activity threads model knowledge of actual attack paths and the interdependence within and between the threads. The nature of activity threads, as defined in this section, allows for an event/vertex to satisfy one or more of the resource requirements of another event (§4.5.6) enabling the later event to occur. Furthermore, each vertex is an event bringing with it the depth of information an event provides making the graph information rich as well as inherently more usable.

Formally, we can define the activity thread as a directed graph, $\boldsymbol{AT}$, where $AT = (V, A)$ is an ordered pair such that:

- $|V| \geq 1$, there exists at least one event in the thread[17]

[16] Attack graphs are an enumeration of all possible paths an adversary may take to penetrate computer networks achieving their desired intent.

[17] While Axiom 4 ensures that there are at least two phases to every activity it is likely that not all are

- AT is a finite graph

- V is the set of all events partitioned into sub-sets such that all events in a sub-set share the same adversary and victim and are further partitioned into p labeled tuples where p is the number of defined phases and each event is placed into the tuple which matches its phase

- A is the set of ordered pairs of arcs such that $arc(x, y)$ is defined if and only if the adversary successfully executed event y because of event x and event x directly preceded event y

- There can exist more than one arc to any one event. For example, given three events x, y, and z there can exist a path from x to y $arc(x, y)$ as well as a path from z to y $arc(z, y)$.

- There can exist more than one arc from any one event. For example, given three events x, y, and z there can exist a path from x to y $arc(x, y)$ as well as a path from x to z $arc(x, z)$.

- There can exist only one path from one node to another (i.e., each arc ordered pair is unique within the graph). For example, given two events x and y there can only exist one path from x to y $arc(x, y)$.

- Arcs are labeled with a 4-tuple $\langle Confidence, And/Or, Hypothetical/Actual, Provides \rangle$ where:

  - Confidence: defining the analytic confidence in the existence of a causal relationship between x and y

  - And/Or: defines whether the path from x to y is necessary and required for y to be successful (AND) or whether the path is an alternative and optional route to achieve y from x (OR)

  - Hypothetical/Actual: distinguishes a hypothesized arc from an actual arc (hypothesis support is described in §8.2) supported by evidence

  - Provides: defining the resources x provides to y to be successful matching with the requirements listed in the resources event meta-feature (§4.5.6)

known at the time of discovery and therefore an activity thread can be initially created with only one event. The empty phase(s) and missing event(s) are then treated as knowledge gaps.

8.1 Adversary Process

Collectively, the vertical threads and horizontal linkages effectively describe the the end-to-end process of an adversary as defined by Axiom 4. This is further enriched by the events themselves which contain the features of the individual actions (e.g., the capability and infrastructure used, the specific methodology, the external resources applied). Together these define *how* the adversary executed their operations, their *modus operandi*.

However, in many cases an adversary will demonstrate a preference for certain elements and behaviors within their broader processes. This fact has been identified and explored in criminology and is likely the outcome of the human attraction to the comfortable and familiar based on culture, knowledge, training, experience, etc. [53] In larger organizations, these preferences will likely also be driven by policies and edicts from leaders. Intrusion analysts usually identify these preferences through common elements across a campaign just as traditional criminal investigators identify them through common evidence amongst crime scenes.

The ability to identify and articulate these common adversary features and behaviors is powerful. With this characterization analysts can group like-threads together which share similar processes (see §9) without the need to match on exact features (e.g., the same infrastructure IP address, the same capability) for every event. The Diamond Model defines this as an *adversary process*.

For example, Figure 7 illustrates an adversary process defined from events 2, 3, 4 and 6 in Figure 6. This adversary process is generally described as: a reconnaissance event which includes a web search for "network administrator", followed (but not necessarily immediately) by the delivery of an email with a trojanized attachment, followed by a specific and known exploit on the local machine (e.g., CVE-YYYY-XXX), and finally an HTTP Post leaving the victim. This thread can now be used to match against other activity threads which exhibit the same general order of events and features.

Formally, adversary processes are defined as sub-graphs of an activity thread which contain a sub-set of their features. Importantly, the sub-graph can be "elastic" in that it can be defined such that events do not need to maintain their strict order to effectively match another thread (illustrated in Figure 7 as dashed arcs between events). In other words, it only matters that the features are matched in the general order but other events can exist between them. Alternatively, an adversary process can be defined "strictly" such that events must maintain their order without intervening events, or a combination of the two.

8.2 Analytic Hypothesis Support

Supporting hypothesis generation, documentation, and testing is one of the most important features of the activity thread and provides for the integration of formal analytic models such as "The Analysis of Competing Hypotheses" (ACH) [1] and applies necessary scientific

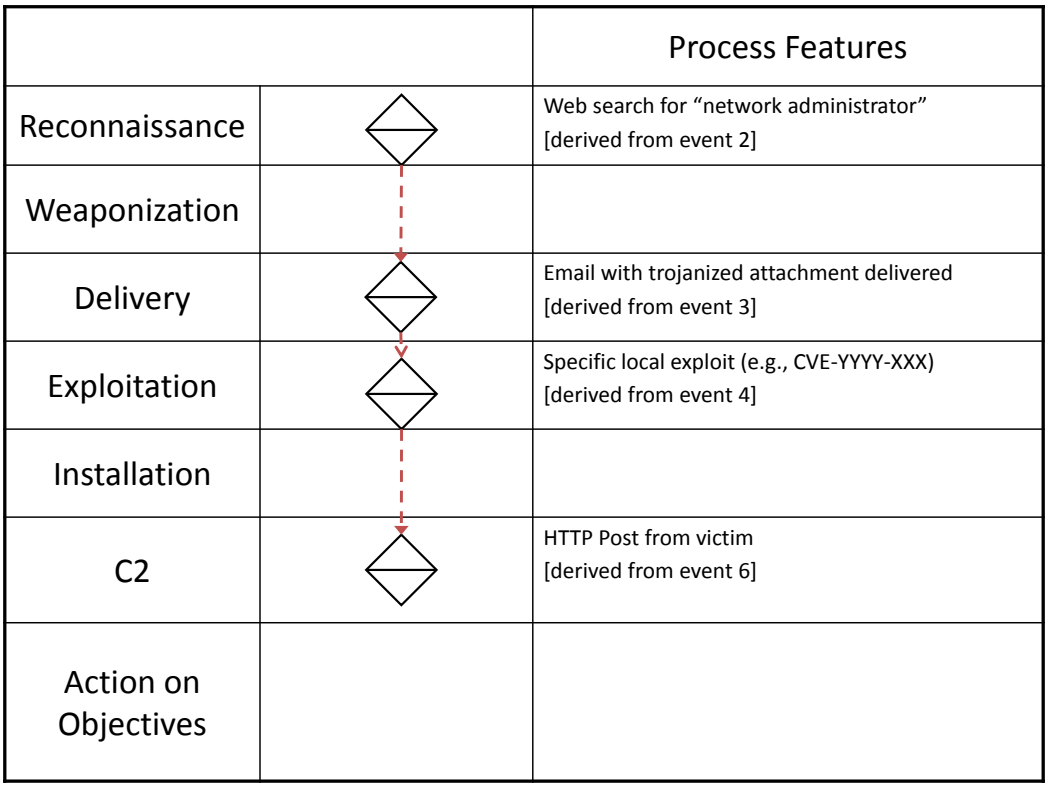

Figure 7: An example adversary process derived from the activity thread illustrated in Figure 6. In this process, features from events 2, 3, 4, and 6 are extracted into a sub-process which can be used to match against other threads. The arcs between events are dashed illustrating that while the events are still phase-ordered other events can intervene between them without disrupting the matching criteria.

rigor. The first step of analysis is to define the question to be addressed. Once the question is defined hypotheses can be generated, documented, and tested.

As described earlier, by placing events within a phase-based model knowledge gaps can be more easily identified. Since Axiom 4 states that malicious activity is multi-phased, each phase *should* contain at least one event.[18] An alternative method of knowledge gap identification is to use the resources meta-feature (§4.5.6). The analyst can then ask how the adversary is fulfilling the required resources for each event generating the necessary hypotheses to address the question.

These hypotheses can then be documented in the activity thread and necessarily differentiated from other events. This is an important feature because one of the failings of most analysis is the lack of documented hypotheses and furthermore, and more dangerously, the lack of differentiation between hypothesis and fact. The activity-thread model encourages hypothesis generation and documentation increasing the value and accuracy of the knowledge.

Once hypotheses are documented and differentiated, they must be refined and tested. There are several methods of hypothesis testing which can be used with our model to determine if a given hypothesis, both itself and amongst others, is reasonable. For instance, one may apply evidence weighting to competing hypotheses [1], Occam's Razor (e.g., simpler explanations are, other things being equal, generally better than complex ones)[19], conservatism (whether the hypothesis 'fits' given other aspects of the activity), and other formal methods of inductive and deductive reasoning to competing hypotheses [1].

For example, Event 10 in Figure 6 could list the following in the resources meta-feature: network access to send email, access to an email account, the target email address, the trojan malware to include in the email, and knowledge of their target to create an email which will bypass filters and entice the target to execute the malware. Neither Event 7 (proxy access) or Event 8 (search results) provide the necessary resources to send an email to the Chief Resource Officer, particularly their email address and role (e.g., knowledge of the target). Therefore, Event 9 is hypothesized as the source of the targeting information enabling the most enticing email to be sent to the correct target.

Event 9 can be tested in several ways. First, it is simple and logical as all of its required resources are met necessitating no further events to be hypothesized. Second, it 'fits' within the capabilities and access of the adversary. Third, evidence can be gathered (e.g., host event logs) to determine if it occurred making the hypothesis measurable and testable to meet scientific rigor.

This form of documentation lends itself to finally achieving repeatability in the intrusion analysis process as other analysts can independently trace the activity-graph establishing

[18] An event in every phase is not guaranteed as Axiom 4 allows for non-essential phases.

[19] In our model, simple can be easily measured by comparing the number of resources an event requires and how many of those are fulfilled given the current events versus having to hypothesize more events to support.

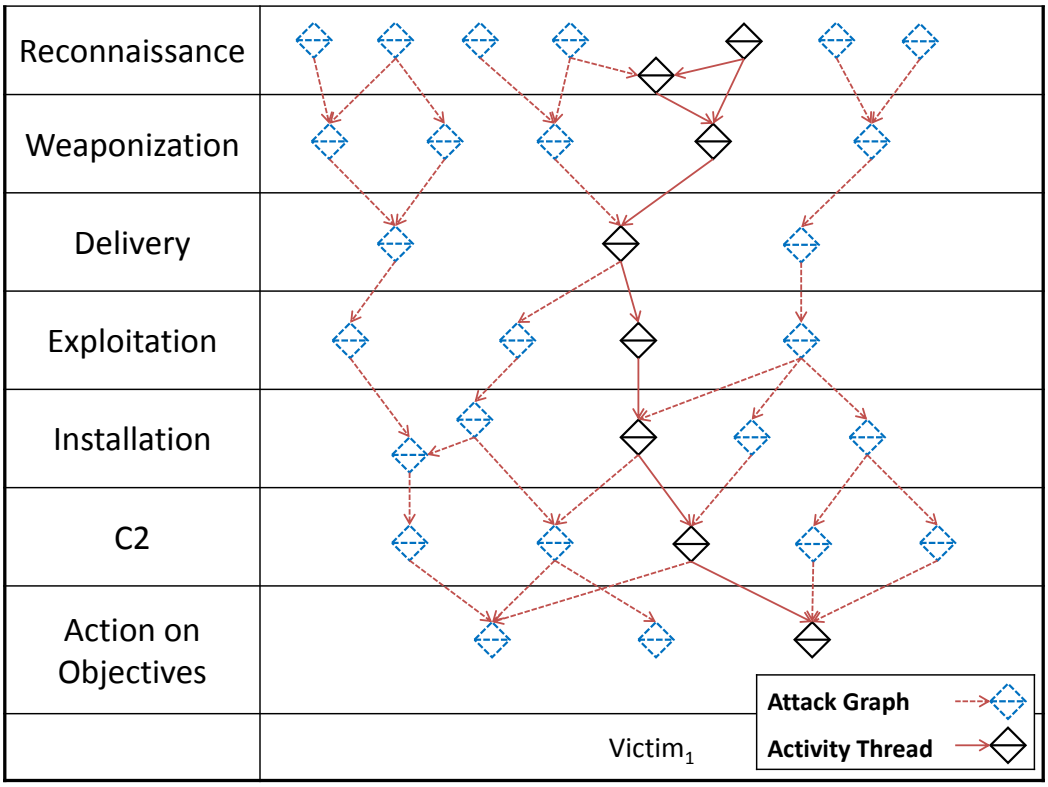

Figure 8: An Activity-Attack Graph example illustrating the integration of knowledge of actual adversary attack paths with the multitude of hypothetical attack paths that could be taken. Using an activity-attack graph highlights the potential paths of an adversary in the future as well as the preferred paths based on current knowledge.

their own hypotheses and conclusions comparing them to the original. This process builds confidence in analytic conclusions and greater accuracy in final judgments.

8.3 Activity-Attack Graph

Activity-Attack Graph Activity threads and traditional attack graphs are not mutually exclusive but instead answer complementary questions. Attack graphs identify and enumerate paths an adversary *could* take while activity threads define the paths an adversary *has* taken. These can exist together by overlaying activity threads on top of a traditional attack graph. We refer to this intelligence-informed attack graph as an *activity-attack graph*.

The activity-attack graph provides several benefits:

- It maintains the integrity of the attack graph making available the full scope of attack graph analysis.

- It increases the amount of information contained in an attack graph as each vertex is a feature-rich Diamond event.

- It increases the amount of visual information contained in the attack graph with little-to-no reduction in usability.

- It generates more accurate weights as actual attacker choices (and preferences) are known.

- It highlights attacker *preferences* alongside alternative paths.

- It exhaustively (due to the nature of attack graphs) maps alternative paths for gaming scenarios and the development of mitigation campaigns (e.g., if this action is taken, the adversary is likely to take one of these paths...).

- It naturally helps fill knowledge gaps for any one attack thread by overlaying the corpus of horizontally related attack threads for comparison. The result is more accurate and faster hypothesis generation and testing during ongoing incident response investigations.

Figure 8.3 is an example of an activity-attack graph. The figure distinguishes the known adversary paths (activity graph) from the possible paths yet known to be exploited (attack graph). This is much like consulting both penetration tester (e.g., Red Team) and vulnerability assessment (e.g. Blue Team) results simultaneously to plot the best course of action.[20]

Ultimately, activity threads and activity-attack graphs enable better mitigation strategy development as they integrate both information assurance and threat intelligence cohesively. They integrate what *has* occurred with what *might* occur allowing a strategy to both counter the current threat and plan for adversary reaction effectively countering future adversary moves. This integrated planning also leads to more efficient resource utilization as mitigation actions can be designed to counter the current threat as well as the future threat simultaneously.

9 Activity Groups

Activity Group An *activity group* is a set of Diamond events and activity threads associated by similarities in their features or processes and weighted by confidence. An activity

[20]Whether the activity-attack graph approach would be a useful method for integrating actual red and blue team results for analysis is not explored, but an interesting question and left for future work.

group has two purposes: (1) a framework to answer analytic questions requiring a breadth of activity knowledge, and (2) the development of mitigation strategies with an intended effect broader than activity threads. Activity groups are differentiated from activity threads in two ways: (1) activity groups contain both events and threads, and (2) events and threads in an activity group are correlated by similar features and behaviors rather than causally related (as is the case with activity threads).

Analysts traditionally form activity groups to identify a common adversary behind events and threads usually using similarities in infrastructure and capabilities. But, the concept is inherently flexible and extends to include any grouping based on similarities to address a multitude of analytic and operational needs. The desired analytic or operational outcome determines the implementation and type of correlation (i.e., grouping function) used. Furthermore, activity groups are not static – just as adversaries are not static. Activity groups must grow and change over time to absorb new knowledge of the adversary including changes in their needs and operations.[21]

There are **six** distinct steps to the process surrounding activity groups:

Step 1: Analytic Problem The particular analytic problem to be solved through grouping

Step 2: Feature Selection The event features and adversary processes used to form the basis of classification and clustering are selected

Step 3: Creation Activity groups are created from the set of events and threads

Step 4: Growth As new events flow into the model, they are classified into the Activity Groups

Step 5: Analysis Activity groups are analyzed to address the analytic problem(s) defined

Step 6: Redefinition Activity groups need to be redefined from time-to-time to maintain their accuracy

Formally, we define an activity group, $\boldsymbol{AG}$ as a set of events and activity threads which share one or more similarities in features or adversary processes:

[21] While clustering and classification are powerful tools which should be encouraged as an analytic force multiplier, they are not without their pitfalls. Clustering and classification schemes have many well-documented concerns. Some are of particular worry because adversaries actively practice denial and deception in almost every packet to evade detection and analysis. The area of most concern is called overfitting, where an analyst or machine includes non-related information into the clusters. The primary reason for this error is poor group definition (i.e., weak feature vector). Particularly in the case of intrusion analysis, it shows itself where two activities overlap features (e.g., sharing a public capability, using shared hosting space). This is further compounded by error propagation: once a non-related event is included in the group definition then it is used for future correlation increasing the number of non-related events compounding the initial error. Several techniques exist to detect and prevent overfitting [54]. A comparison of these techniques as applied to intrusion analysis is beyond the scope of this work and is left for future work. However, it is a problem worth noting.

$$AG = \{et_1, et_2, \ldots, et_n\}$$

Where:

- $n \geq 1$, there must be at least one element in an activity group
- et_n is either: A singular event or an activity thread as defined in §8
- All events or processes in AG share one or more similarities satisfying the activity group creation function used to partition the events and threads (defined in §9.3)

9.1 Step 1: Analytic Problem

Activity grouping is used to solve a number of problems. These problems generally require deduction and inference based on a common set of features (i.e., feature vector). These problems are generally distinct enough to require a different feature vector for each problem.[22] For instance, the feature vector which would group events and threads by likely adversary (e.g., attribution) would not always suffice to group events to discover common malware authors/developers. The analytic problem must first be defined.

Therefore, we define an analytic problem, ***PR***, as an intrusion analysis problem statement which requires clustering and classification (i.e., grouping) to address in part or full.

Some examples of analytic problems which activity groups support:

- Trending: How has an adversary's activity changed over time and what is the current vector to infer future change?
- Intent Deduction: What is the intent of the adversary?
- Attribution Deduction: Which events and threads are likely conducted by the same adversary?
- Adversary Capabilities and Infrastructure: What is the complete set of observed capabilities and infrastructure of the adversary?
- Cross-Capability Identification: Which capabilities have been used by multiple adversaries?
- Adversary Campaign Knowledge Gap Identification: What are the organization's knowledge gaps across an adversary's campaign?

[22] However, this does not rule out the possibility that two or more problems share a common feature vector.

- Automated Mitigation Recommendation: When an event is detected which adversary is behind the event and what action can/should be taken? [23]

- Common Capability Development Deduction: Which capabilities show evidence of common authors/developers?

- Center of Gravity Identification: Which resources and processes are the most common and critical to an activity and/or campaign?

9.2 Step 2: Feature Selection

Diamond events and threads are correlated and grouped in two complementary ways: (1) using various event core-, meta-, and sub-features (e.g., infrastructure, capability), and (2) adversary processes (§8) previously defined as activity group sub-graphs. To accomplish this, features are selected populating a feature vector defining the elements used to group events and threads.

Importantly, feature vectors can be tremendously specific allowing an analyst to define a particular activity of interest by including the particular observables (e.g., IP addresses, domains, malware) such that two groups are formed: events and threads which are part of the activity, and those which are not.

Furthermore, processes included in a feature vector are a powerful concept to support comparing activity not only by observable but also by specific means irregardless of specific infrastructure or capability. This is especially effective against adversaries who may change infrastructure and capability (the most common observables) often but maintain a semi-static process.

The *feature space* is composed of all core- and meta-features of the events (e.g., infrastructure, capability, victim, result) (§4) as well as any defined adversary processes (§8). From the feature space the most relevant and optimal features are selected and/or created[24] defining the *feature vector*. Lastly, each of these features can be combined with a weight identifying its relative importance in defining the group. Several well-known techniques exist to select (and possibly create) the most relevant and optimal features [55]. Further discussion of optimal Diamond Model feature selection/creation for activity grouping is left as an area for future research and will also be implementation-specific.

[23] Activity groups support real-time intelligence-driven network defense. As events are detected in real-time, machine learning classification techniques are applied classifying and associating events known activity groups. Given a set of pre-established conditions (e.g., if event E is classified as activity group X with $> 80\%$ confidence) the system can make a recommendation to network defense mechanisms to apply mitigation techniques to the activity. In this way, adversary operations can be mitigated in real-time even if the adversary has changed part of their operations without requiring defenders to forecast the change.

[24] *Feature creation* (the creation of new features from existing features) is noted because often analysts use a function to compare features based on intrusion activity. For example, the IP address of a resolved Domain Name may be an extracted feature if the IP did not exist in the original feature list now allowing events referencing the same Domain OR its associated IP to be correlated.

Table 3: Example Activity Group Definition Steps 1 and 2

Analytic Problem	Which events and threads are likely to be conducted by the same adversary who utilizes a certain process ($Process_1$)? (e.g., attribution)
Feature Space	$Infrastructure_{IP}$, $Infrastructure_{Domain}$, $Capability_{MD5}$, $Victim_{IP}$, $Victim_{Organization}$, Methodology, $Process_1$, $Process_2$, $Process_3$
Feature Vector	$\langle$Infrastructure$_{IP}$, $Capability_{MD5}$, $Process_1$ $\rangle$
Outcome	All events and threads will be grouped by similarities in infrastructure IP, capability MD5 hash, and a defined adverarsy process $Process_1$

We define the feature space, ***FS***, as the set of all core-, meta-, and sub-features which define events as well as any and all adversary processes.

Further, we define the feature vector to address an analytic problem, FV_{PR}, as

$$FV_{PR} = \langle\langle f_1, w_{f_1}\rangle, \langle f_2, w_{f_2}\rangle, \ldots, \langle f_n, w_{f_n}\rangle\rangle$$

Where:

- $n \geq 1$, there must be at least one element in the feature vector
- $f_n \in FS$, every feature in the feature vector must exist in the feature space
- $FV \subset FS$, the feature vector is a sub-set of the feature space
- f_n is a necessary element to group events and threads to address the analytic problem PR
- $w_{f_n} \in \mathbb{R}$ and $0 < w_{f_n} \geq 1$, the weight is a real number which describes the relative importance of f_n to all other f, such that $w = 1$ is a feature with the greatest importance[25]

[25]There should be no feature with a given weight of zero in the feature vector as that would indicate the feature had no importance. In that case the feature should not be included in the feature vector.

9.3 Step 3: Creation

Analysts initially create activity groups through a cognitive clustering process: an analyst compares an event's features with every other (e.g., their feature vector) and using some function defining similarity separates the events into distinct groups (i.e., sets) with an associated confidence of which group the event belongs. Formally, these groups become *classes* upon which machine learning techniques can be applied, such as classification in activity group growth (§9.4).

It is expected that an organization will have more than one analytic problem defined (via Step 1). Therefore, there is likely more than one activity group creation function per Diamond Model instance. For example, grouping events by apparent same actor-adversary (i.e., grouping for attribution) and grouping events by victim vulnerability (e.g., grouping for most likely exploitation path) are different analytic problems requiring distinct functions. Furthermore, some problems may require the clustering function to place every event and thread into a group where others may allow outliers (i.e., events and threads which belong to no group).

Activity group creation is a general clustering problem solving event and thread correlation addressing a particular analysis problem. The clustering function is dependent on prior information such as the analytic problem/objective and the particular feature vector which can be unique for any given application of the Diamond Model. Therefore, it is likely that there is no one activity group creation function to solve all intrusion analysis problems. We expect further research defining functions optimizing clustering for particular intrusion analysis problems, such as optimized event clustering for adversary attribution.

Formally, we define an activity group creation function, $\boldsymbol{AGC}$ as:

$$AGC(PR, FV_{PR}, ET) \rightarrow AGS$$

$$AGS = \{AG_1, AG_2, \ldots, AG_n\}$$

Where:

- PR is a defined analytic problem to be satisfied by the function
- FV_{PR} is the feature vector which satisfies the analytic problem PR
- ET is the set of all events and threads to be grouped
- AGC partitions all elements of the event/thread set ET into a set of n Activity Groups, AGS, based on the feature vector FV_{PR}

- The function comprising AGC can operate across all elements within the set ET using the features and processes defined in FV_{PR}[26]

- AGS is the set of activity groups such that each activity group, AG_n, satisfies the definition of an activity group

- It is possible that the creation function establishes no groups because no similarities exist, and therefore $n \geq 0$

9.3.1 Activity Group Creation Example

Figure 9 illustrates how an activity group creation function (AGC), using strict partitioning with outliers, can be defined to answer the example problem posed in the previous Feature Vector section (§9.2) defining groups based on a common adversary likely to utilize the same infrastructure IP, capability, and a 3-step process. The illustration shows a nominal set of 17 events and threads (ES) which are grouped according to our function and feature vector to create three groups and two errant events and one errant thread which did not meet the function's criteria and are left un-grouped. Our function grouped two threads into Activity Group 3.

For our example, we'll say that the logic expressed in the function states that any thread which contained the process $A \rightarrow B \rightarrow C$ would be an Activity Group. Two or more process-matching threads would be correlated within the same activity group if at least one event within each thread (not necessarily within the specified process) shared an infrastructure IP and capability MD5 hash with at least medium confidence. Now that the data is organized to answer the analytic question, the groups can be grown (§9.4) and analyzed (§9.5) to provide insight potentially answering the question.

9.4 Step 4: Growth

Analysts continuously grow activity groups through a cognitive pattern recognition process mimicking confidence-weighted probabilistic classification: an analyst discovers a malicious event, compares the event to all other known events based on feature similarities and their confidence, and associates (i.e., classifies) the event with the most similar group (i.e., class) (or alternatively abstains from association if the confidence does not meet their threshold). This action continuously grows the activity groups as events and threads are characterized into the groups as they are discovered, detected, or received.

Figure 10 illustrates activity group growth: as events and threads are discovered, detected, or received they are classified into the various activity groups based on the defined feature

[26]The operations in an activity group creation function are as broad as necessary and are not required to use all elements of the feature vector.

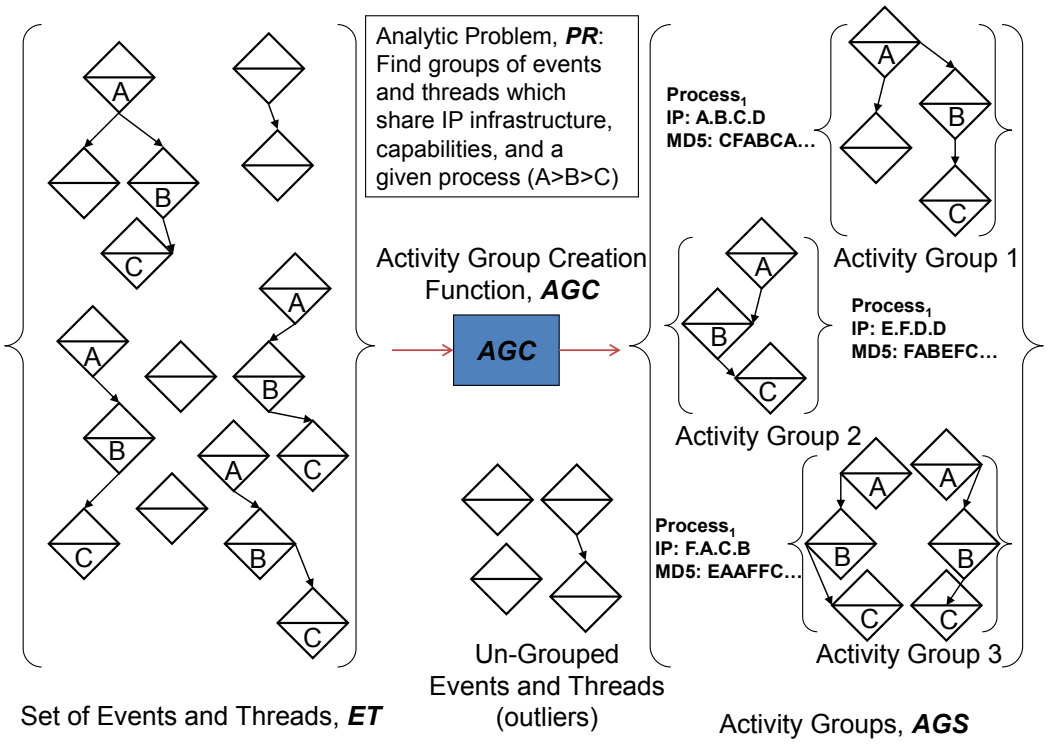

Figure 9: Activity group creation is illustrated such that a group of events and threads are clustered based on a feature vector defined by: an adversary process ($A \rightarrow B \rightarrow C$), a matching capability MD5 hash, and infrastructure IP address. Based on this feature vector and an activity group creation function (AGC) the 17 events and threads are clustered into three groups with two events and a thread not meeting the grouping criteria and categorized as outliers.

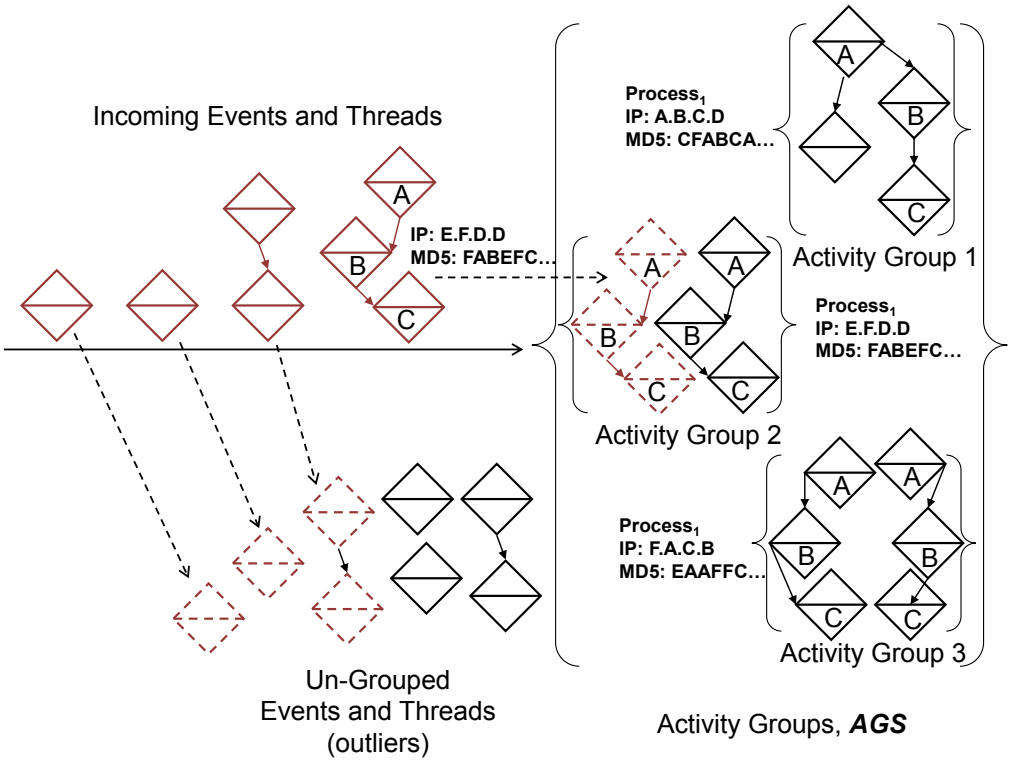

Figure 10: Illustrating Step 4, Activity Group Growth, events and threads are discovered, detected, or received and continuously classified into existing activity groups based on the previously defined feature vector. In this instance, those events and threads not successfully matching the criteria are outliers and not grouped.

vector. In this example, the thread meeting the criteria is classified into Activity Group 2 while the other thread and events are classified as outliers.[27]

9.5 Step 5: Analysis

Once an activity group is defined and events and threads are clustered within the groups it can be analyzed to address the specific analytic problem being addressed. This generally requires the application of tools and tradecraft beyond the Diamond Model. For instance, with our example in Figure 9 the analyst will now likely examine each of these groups to discern differences and similarities exposing new analytic problems to be solved. This may even lead to a re-examination of the feature selection and grouping function requiring redefinition (the next step).

Nevertheless, the analyst now has the tools to analyze intrusion events and threads across a larger scale including: potentially exposing longer-range adversary campaigns, identifying similarities between seemingly dissimilar events, gathering a complete listing of observed adversary capabilities and infrastructure, deducing adversary attribution based on the victim-set (i.e., cyber-victimology §5.1.2), and many other problems.

9.6 Step 6: Redefinition

Activity groups, like all clustering and classification-based functions, suffer from various well-studied challenges. One such challenge is the assumption that the analyst can accurately describe the feature vector and function used to cluster – or that their idea of a cluster is correct to begin with. Another is overfitting and error propagation: where an analyst or a system wrongly associates an event to a group propagating and potentially magnifying that mistake over time. Therefore, it is normal that activity groups require examination, anomaly detection, and redefinition (re-clustering) over time to discover and correct errors. Furthermore, during this redefinition stage changes can (and should) be considered to the feature vector and associated weights and algorithms to ensure the underlying error is corrected. Manually, this is usually done through the discovery of evidence which indicates an incorrect classification has occurred requiring re-clustering.

9.7 Activity Group Families

Activity groups are as varied as the enterprises behind the malicious activity. As such, the identification and detection of millions of events could easily filter into a large number of

[27] However, as previously described in Step 3, Activity Group Creation (§9.3), the clustering function is defined by the needs of the analyst and the particular analytic problem being solved and therefore alternative clustering types are possible which may not use outliers but instead place every thread and event into a group.

activity groups, some of which interact at a higher-level. Therefore, it is sometimes necessary to develop a hierarchy of groups which model the increasingly complex organizations behind the events in order to address higher-order questions and develop even more strategic mitigation.[28]

Very much like an activity group, an activity group family is a set of activity groups which share common features, except that the common features of groups within a family are likely non-technical. For example, in the case of organized crime, a common funding and tasking element may be responsible for multiple operations and therefore multiple activity groups – each of which are tracked and analyzed separately – are grouped within a family. This makes the identification, organization, and mitigation strategy development of higher-order elements, such as the crime-boss in this example, tractable and more effective.

For the purposes of analytic methodology, activity group families are treated to the same 6-step process as an activity group. They must be defined, created, grown, analyzed, and redefined. They also have feature vectors and creation functions except that the creation function used for clustering and classification operates across the features of an entire group rather than individual events or threads. These terms and the associated functions, features, and processes do not need to be re-defined as they have been discussed in full previously – except to say that they are slightly modified to support features and processes across activity groups.

Formally, we define an activity group family as:

$$AGF = \{AG_1, AG_2, \ldots, AG_n\}$$

Where:

- $n \geq 1$, an activity group family must contain at least one activity group
- AG_n satisfies the definition of an activity group
- AGF is a set of activity groups which share one or more similarities
- AGF satisfies a particular analytic problem
- AGF is the outcome of a creation function and feature vector comparing activity groups

[28] Evidence of such organization behind malicious activity is apparent in the "Phonemasters" case [45] and Brenner argues persuasively in [56] that existing hierarchical organized crime models will necessarily follow into cyberspace as they are the most efficient method of varied criminal enterprises – and cyberspace will be a natural extension of criminal activities, especially the largest ones. However, our model is not limited to cybercrime but extensible to any organized enterprise conducting a multitude of malicious cyber activities which are necessary to group.

10 Planning and Gaming

From a mitigation perspective, many actions are possible. However, deciding the best action to take to offset the adversary is challenging. Actions cost defenders money and/or time to implement and are taken with the expectation that the action will adversely impact adversarial efforts.

Our model provides an understanding of dependencies between adversary components. For adversary efforts to succeed, complete threads must be available creating a pathway between the intent and result. The model aids in understanding how defender actions will impact adversary capabilities by determining which components an adversary will need to replace/fix/re-implement.

Additionally, defender actions should be chosen that cost defenders little but cost adversaries much more. Clearly, the reverse (i.e., costing the defender more and the adversary less) is undesirable from a tactical or strategic mitigation perspective. Actions that cost the defender more (especially significantly more) should be avoided if at all possible. The cost to the adversary can be expressed as the cost (in money, resources, time) to recoup the necessary capability and infrastructure to have a functional platform. Adversary cost has multiple components including development time, infrastructure building cost/time, retraining time and costs, opportunity cost, and costs incurred from loss of readiness. Defender costs also have multiple components such as money, time, as well as legal and ethical risks which need to be addressed [13].

The Diamond Model is a fundamental concept helping to structure and strengthen analysis to achieve its ultimate objective: mitigation. The Model does not prescribe mitigation strategy or course of action development. These exist separate from the Model. Instead, it supports many forms of decision making. The following are discussions on the model's applicability to aspects of several popular decision frameworks:

Joint Intelligence Preparation of the Operational Environment (JIPOE) US Department of Defense military planning doctrine Joint Intelligence Preparation of the Operational Environment (JIOPE) [12] is a well-understood and often cited resource which establishes a process for the use of intelligence to develop courses of action. The doctrine recognizes that a strategy will fail if based on only the nullification of adversary infrastructure and capabilities. Rather, it prescribes a combined approach which also includes the identification of adversary resources, centers of gravity, as well as adversary responses and courses of action. This approach identifies optimal areas for mitigation and counters the adversary's capacity to maintain and rebuild those capabilities and infrastructure once mitigated. Our model supports such planning in that it:

- Assists in the identification of intelligence and information gaps through missing event features and phase-gaps in activity threads (JIOPE Step 1, Element 6)

- Supports the development of an adversary model (JIOPE Step 3, Element 1)
- Identifies adversary infrastructure and capabilities with a focus on resources (JIOPE Step 3, Element 3)
- Identifies adversary centers of gravity through activity thread and activity group analysis (JIOPE Step 3, Element 4)
- Identifies adversary objectives and end state through activity thread analysis, victimology, and activity groups (JIOPE Step 4, Element 1)
- Determines likely adversary courses of action through activity-attack graph analysis where potential and preferred paths of attack can be identified (JIOPE Step 4, Elements 2 and 3)

Kill Chain Analysis The Diamond Model and Kill Chain analysis are highly complementary. Kill Chain analysis allows an analyst "to target and engage an adversary to create desired effects." [11] The Diamond allows analysts to develop tradecraft and understanding in order to build and organize the knowledge necessary to execute the Kill Chain analysis. Two methods of integrating the two approaches are described:

- Once an analyst develops an activity thread, courses of action for each event along the thread can be identified using the Kill Chain's course of action matrix. As illustrated in Figure 11, courses of action for each of the Kill Chain stages are identified for activity threads 1 & 2 from Figure 6. The power of the Diamond Model is that courses of action can be designed to span multiple victims and across the activity of an adversary making the actions even more powerful as they further reduce the capacity of the adversary.
- Activity groups clustered by same likely adversary (i.e., clustering by attribution) with analysis of the largest common feature set amongst the events in a group can provide the Kill Chain's required key campaign indicators necessary to focus and prioritize courses of actions.

Vulnerability Cover Common information assurance practice is to analyze a system (or network) for vulnerabilities, ranking those vulnerabilities based on the specific concerns of the organization (e.g., asset value and cost), followed by the application of mitigation to those vulnerabilities. Through the generation of activity-attack graphs, this normal process of deciding which paths of the graph to prune by mitigation (thereby denying use of the path by the adversary) is now informed by *adversary preference* and *potential*. Through the use of our model, traditional information assurance decisions are no longer made by hypothesizing potential adversaries and their paths but rather by injecting actual attack paths into the graph as well as projecting adversary preference and potential. This approach provides more complete protection and increases the adversary cost as they must now develop, train, and

	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance	Web Analytics	Policy to Prevent Forum Use			Create fake postings	
Weaponization						
Delivery	NIDS, User Education	Email AV Scanning		Email Queuing	Filter but respond with out-of-office message	
Exploitation	HIDS	Patch	DEP			
Installation						
C2	NIDS	HTTP Whitelist	NIPS	HTTP Throttling		
Action on Objectives	Proxy Detection	Firewall ACL	NIPS	HTTP Throttling	Honeypot	

Figure 11: Kill Chain Course of Action Matrix derived from threads 1 & 2 in Figure 6. Mitigation actions for each category (e.g., disrupt, degrade, deny) were identified to counter the effectiveness of the adversary's events along the phases. This matrix format and process is described in [11] as a method of identifying mitigation courses of action to counter adversary campaigns – illustrating the power of combining the Diamond Model and Kill Chain analysis.

operate beyond their current capability base.

Gaming Part of any effective mitigation strategy development is to game the adversary in order to predict their next move. In this way a defender can both counter the current activity and pre-position for future activity thereby countering the adversary. Planning for both at the same time enables economical decisions satisfying both requirements. Our model supports gaming in several ways to more accurately predict adversary responses to environmental pressures (e.g., defender actions, patched vulnerability):

- It enables higher-order gaming around human decision making as the social-political and intent meta-features are an integrated aspect (e.g., what are the needs and aspirations of the adversary? How can those be influenced or countered?).
- Through activity threads and groups, attribution deduction is possible via cyber-victimology (§5.1.2) and other means.
- The fundamental support of hypothesis testing enables a more complete gaming scenario improving the value of gaming and the accuracy of its outcome.

11 Future Work

We recognize that the Diamond Model at present is cognitive and highly manual. We are willing to accept this because it is, as its name implies, a model to be studied and refined defined with respect only to accurately capturing the intrusion analysis process. However, we are ultimately pragmatists and recognize that the model will be many times more useful once automation and efficiencies are developed. As such, we hope we have provided sufficient insight and citations of related efforts to motivate future work along these lines.

One such invaluable automation would be the integration of the Diamond Model into analytic tools which are both automatically fed with intelligence from network sensors as well as external reports from other organizations, especially those within a shared threat space (§5.1.3). However, we expect that intrusion analysts will still be required to input new intelligence and oversee automated feeds. This will require work into usability to augment, rather than impede, the analytic work-flow. Furthermore, this could also bring about automated and formal hypothesis generation and testing providing instant evaluation of analytic conclusions.

In order to achieve this, there must be a protocol to share contextual indicators and threat intelligence to rapidly integrate information from all of these sources. We see the Diamond Model as a foundation for achieving this and improving new or existing protocols and formal languages (as [30, 27, 22, 24, 32]) to make them more contextual and relational. This will also likely require further refinement of taxonomies. The Diamond Model itself

provides the opportunity to define the features and sub-features into a never-ending strata of information. However, different implementations of the model could conflict in their definitions. Therefore, further refinement of the sub-models for each feature and sub-feature using taxonomy fundamentals is critical [25].

There are also several miscellaneous elements which have been described as necessitating future effort, such as:

- The definition of feature vectors and clustering/classification algorithms for particular analytic problems
- The potential integration of penetration test and vulnerability assessment output into activity-attack graphs
- Methods of preventing overfitting of intrusion analysis events during clustering/classification
- A thorough examination and definition of event sub-features as a taxonomy
- The evaluation of variables and aspects to determining degrees of persistence
- A more thorough understanding of the Social-Political sphere and its role in mitigation decision making, including accounting for adversary needs and aspirations

Lastly, the purpose of the model is to achieve more effective and accurate analysis ultimately to enable planning, strategy, and decision-making to defend networks. Therefore, while we have shown how the model can be used with several planning frameworks, each one could be a work in itself and there are many other models to consider.

For instance, a potential path to generate more effective and creative mitigation strategies would be to extend the work of [14] and treat the activity thread as a group of chromosomes in a co-evolutionary predator-prey environment using genetic algorithms. This approach has previously shown promise and the activity threads model a well-behaved genetic algorithm chromosome lending credence to this concept.

12 Conclusion

This paper presented the Diamond Model of intrusion analysis. It began with the atomic element of all intrusion activity, the event, and its core features (adversary, victim, infrastructure, and capability) organized in the shape of a diamond. This event was further refined with sub-features and meta-features allowing it to contain and relate all aspects of a malicious event. From the event we derived several feature-centric approaches to assist in the categorization of existing analytic tradecraft and the development of new tradecraft. The model further extracted new understandings about malicious activity such as the importance of the Social-Political relationship between adversary and victim and the degrees

of persistence.

Furthermore, the Diamond Model captured the essence of intrusion activity as a set of causal events related in an activity thread documenting the end-to-end process of the adversary. Importantly, these threads are further augmented with attack graphs to create a new intelligence-driven approach to traditional information assurance called activity-attack graphs taking into account actual adversary attacks as well as potential and preferred paths. The threads and events are then coalesced into activity groups which address broader analytic problems and enable more strategic mitigation campaigns to be developed. Lastly, activity groups can be hierarchical and organized into families which better model sophisticated adversary organizations.

The Model has also been shown to be highly complementary with multiple mitigation planning and decision models including the Joint Intelligence Preparation of the Battlespace, the Kill Chain, traditional information assurance vulnerability coverage, and adversary gaming approaches.

Intrusion analysis has long been regarded as an art to be learned and practiced, rather than a science to be studied and refined. Evidence of this is everywhere: from the focus on analytic outcomes more than process and principles, to the transmission of knowledge via stories and case-studies. However, approaching it only as an art has long delayed improvements and understanding further slowing the evolution of threat mitigation which relies on efficient, effective, and accurate analysis. Without knowing it, analysts have used the Diamond Model for decades but have lacked the complete framework to understand, improve, and focus their efforts.

It is time to recognize that the discipline is both an art and a science. The Diamond Model addresses this challenge head-on integrating the art and science of intrusion analysis. The Diamond Model accurately captures and organizes the foundational and fundamental concepts which underpin all that intrusion analysts do as well as how intrusion analysis is synthesized and used for mitigation and network defense. It has achieved its aims of being both an informal cognitive analytic support and a formal framework applying mathematical and computational concepts to intrusion analysis. However, its largest contribution is that it finally applies the scientific rigor and the principles of measurement, testability, and repeatability to the domain enabling intrusion analysis to become more effective, efficient, and accurate leading to quicker, more effective, and more efficient mitigation to defeat our adversaries.

References

[1] Richards J. Heuer Jr. *Psychology of Intelligence Analysis.* Central Intelligence Agency, 1999.

[2] Chris Sanders. The 10 commandments of intrusion analysis. [ONLINE] `http:`

//chrissanders.org/2011/01/the-10-commandments-of-intrusion-analysis/, January 2011.

[3] Leo Obrsta, Penny Chaseb, and Richard Markeloffa. Developing an ontology of the cyber security domain. In Paulo C. G. Costa and Kathryn B. Laskey, editors, *Proceedings of Semantic Technologies for Intelligence, Defense, and Security (STIDS) 2012*, pages 49–56, October 2012. [ONLINE] `http://ceur-ws.org/Vol-966/`.

[4] Clifford Stoll. Stalking the wily hacker. *Communications of the ACM*, 31(5):484–497, May 1988.

[5] Steve Bellovin. There be dragons. In *3rd Usenix UNIX Security Symposium*, Baltimore, MD, USA, September 1992.

[6] Bill Cheswick. An evening with Berferd. In *Firewalls & Internet Security*, chapter 10. Addison-Wesley, Reading, MA, USA, 1994.

[7] Lance Spitzer. The honeynet project: Trapping the hackers. *Security & Privacy*, page 15, April 2003.

[8] Stephen Northcutt, Mark Cooper, Matt Fearnow, and Karen Frederick. *Intrusion Signatures and Analysis*. New Riders Publishing, Indianapolis, IN, USA, 2001.

[9] SANS. [ONLINE] `http://www.sans.org`.

[10] Bernhard Amann, Robin Sommer, Aashish Sharma, and Seth Hall. A lone wolf no more: Supporting network intrusion detection with real-time intelligence. In *15th International Conference on Research in Attacks, Intrusions, and Defenses*, pages 314–333, Berlin, Heidelberg, 2012. Springer-Verlag.

[11] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In L. Armistad, editor, *International Conference on Information Warfare and Security*, volume 6, pages 113–125. Academic Conferences International, Academic Publishing International Unlimited, 2011.

[12] US Department of Defense. *Joint Tactics, Techniques, and Procedures for Joint Intelligence Preparation of the Operational Environment (JP 2-01.3)*, June 2009.

[13] Sergio Caltagirone and Deborah Frincke. ADAM: Active defense algorithm and model. In N.R. Wyler and G. Byrne, editors, *Aggressive Network Self-Defense*, pages 287–311. Syngress Publishing, Rockville, MD, USA, 2005.

[14] Sergio Caltagirone. Evolving active defense strategies. Technical Report CSDS-DF-TR-05-07, University of Idaho, Moscow, ID, USA, 2005.

[15] Bruce Schneier. Attack trees. *Dr. Dobbs Journal*, 24(12):21–29, 1999.

[16] Richard Paul Lippmann and Kyle William Ingols. An annotated review of past papers on attack graphs. Technical Report PR-IA-1, Massachusetts Institute of Technology, Lexington Lincoln Laboratory, 2005.

[17] Xinming Ou, Wayne Boyer, and Miles McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 336–345. ACM, 2006.

[18] Kyle Ingols, Richard Lippmann, and Keith Piwowarski. Practical attack graph generation for network defense. In *22nd Annual Computer Security Applications Conference, ACSAC'06*, pages 121–130. IEEE, 2006.

[19] Lingyu Wang, Tania Islam, Tao Long, Anoop Singhal, and Sushil Jajodia. An attack graph-based probabilistic security measure. In *Data and Applications Security XXII*, pages 283–296. Springer, Berlin Heidelberg, 2008.

[20] John Homer, Ashok Varikuti, Xinming Ou, and Miles McQueen. Improving attack graph visualization through data reduction and attack grouping. In *Visualization for Computer Security*, pages 68–79. Springer, Berlin Heidelberg, 2008.

[21] Sebastian Roschke, Feng Gheng, and Christopher Meinel. Using vulnerability information and attack graphs for intrusion detection. In *Proceedings of the 6th International Conference on Information Assurance and Security (IAS 2010)*, pages 104 – 109, Atlanta, GA, USA, 2010. IEEE Press.

[22] Vocabulary for event recording and incident sharing (VERIS). [ONLINE] `http://www.veriscommunity.net`.

[23] ThreatConnect. [ONLINE] `http://www.threatconnect.com`.

[24] A structured language for cyber threat intelligence information (STIX). [ONLINE] `http://stix.mitre.org`.

[25] John D. Howard and Pascal Meunier. Using a common language for computer security incident information. In Seymour Bosworth and M.E. Kabay, editors, *Computer Security Handbook*, chapter 3, pages 3.1–3.22. John Wiley & Sons, New York, NY, USA, 4th edition, 2002.

[26] Frederick B. Cohen. *Protection and Security on the Information Superhighway*. John Wiley & Sons, New York, NY, USA, 1995.

[27] Steven T. Eckmann, Giovanni Vigna, and Richard A. Kemmerer. STATL: An attack language for state-based intrusion detection. *Journal of Computer Security*, 10(1):71–163, 2002.

[28] Frederick B. Cohen. Information system attacks: A preliminary classification scheme. *Computers and Security*, 16(1):29–46, 1997.

[29] John D. Howard and Thomas A. Longstaff. A common lanaguage for computer security incidents. Technical Report SAND98-8667, Sandia National Laboratories, October 1998.

[30] Frederic Cuppens and Rodolphe Ortalo. LAMBDA: A language to model a database for detection of attacks. In *Proceedings of the Third International Workshop, RAID 2000*, pages 197–216, Berlin, Heidelberg, 2000. Springer-Verlag.

[31] Michel Cedric and Ludovic Me. ADELE: An attack description language for knowledge-based intrusion detection. In *Trusted Information*, number 65, pages 353–368. Springer US, 2002.

[32] Sophisticated indicators for the modern threat landscape: An introduction to OpenIOC. [ONLINE] `http://openioc.org/resources/An_Introduction_to_OpenIOC.pdf`.

[33] Command and control. In *Joint Publication 1-02: Department of Defense Dictionary of Military and Associated Terms*, page 103. US Department of Defense, March 2009.

[34] Wim Van Eck. Electromagnetic radiation from video display units: An evesdropping risk? *Computers & Security*, 4(4):269–286, 1985.

[35] MITRE. Common vulnerabilities and exposures. [ONLINE] `http://cve.mitre.org/`.

[36] Stuart McClure, Joel Scambray, and George Kurtz. *Hacking Exposed*. McGraw-Hill Osborne Media, 4th edition, 2003.

[37] classtype. In *Snort Users Manual 2.9.3*, page 179. The Snort Project, May 2012.

[38] Austin Troya, J. Morgan Groveb, and Jarlath O'Neil-Dunne. The relationship between tree canopy and crime rates across an urban-rural gradient in the greater Baltimore region. *Landscape and Urban Planning*, 106(3):262–270, June 2012.

[39] Will Gragido. Lions at the watering hole – the "VOHO" affair. [ONLINE] `http://blogs.rsa.com/lions-at-the-watering-hole-the-voho-affair`, July 2012.

[40] Kurt Baumgartner. Winnti-stolen digital certificates re-used in current watering hole attacks on Tibetan and Uyghur groups. [ONLINE] `http://www.securelist.com/en/blog/208194218/Winnti_Stolen_Digital_Certificates_Re_Used_in_Current_Watering_Hole_Attacks_on_Tibetan_and_Uyghur_Groups`, April 2013.

[41] Matthias Vallentin, Jon Whiteaker, and Yahel Ben-David. The gh0st in the shell: Network security in the Himalayas. [ONLINE] `http://www.eecs.berkeley.edu/~yahel/`

papers/network-security-in-the-himalayas-cs294-28.pdf.

[42] Symantec Security Response. W32.Duqu: The precursor to the next Stuxnet. [ONLINE] http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_duqu_the_precursor_to_the_next_stuxnet.pdf, November 2011.

[43] Red October: Diplomatic cyber attacks investigation. [ONLINE] http://www.securelist.com/en/analysis/204792262/Red_October_Cyber_Attacks_Investigation, 2013.

[44] Command Five Pty Ltd. SK Hack by an advanced persistent threat. [ONLINE] http://www.commandfive.com/papers/C5_APT_SKHack.pdf, September 2011.

[45] D. Ian Hopper and Richard Stenger. Large-scale phone invasion goes unnoticed by all but FBI. *CNN*, December 1999. [ONLINE] http://edition.cnn.com/1999/TECH/computing/12/14/phone.hacking/.

[46] Nate Anderson. How one man tracked down Anonymous – and paid a heavy price. *Ars Technica*, February 2011. [ONLINE] http://www.arstechnica.com/tech-policy/2011/02/how-one-security-firm-tracked-anonymousand-paid-a-heavy-price/.

[47] Jose Nazario. Georgia DDoS attacks – a quick summary of observations. [ONLINE] http://ddos.arbornetworks.com/2008/08/georgia-ddos-attacks-a-quick-summary-of-observations, August 2008.

[48] Brian Krebs. Cyber attacks target pro-Tibet groups. *Washington Post*, March 2008. [ONLINE] http://www.washingtonpost.com/wp-dyn/content/article/2008/03/21/AR2008032102605.html.

[49] Bojan Zdrnja, Nevil Brownlee, and Duane Wessels. Passive monitoring of DNS anomalies. In *Proceedings of the 4th International Conference on Detection of Intrusions and Malware and Vulnerability Assessments*, DIMVA '07, pages 129–139, Berlin, Heidelberg, 2007. Springer-Verlag.

[50] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolas Vasiloglou, II, and David Dagon. Detecting malware domains at the upper DNS hierarchy. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 27–27, Berkeley, CA, USA, 2011. USENIX Association.

[51] Wolfgang John and Tomas Olovsson. Detection of malicious traffic on back-bone links via packet header analysis. *Campus-Wide Information Systems*, (25):342–358, 2008.

[52] S. Templeton and K. Levitt. A requires/provides model for computer attacks. In *Proceedings of the 2000 Workshop on New Security Paradigms*, New York, NY, USA, 2001. ACM Press.

[53] Crime pattern analysis: An investigative tool. In Michael J Palmiotto, editor, *Critical Issues in Criminal Investigation*, pages 59–69. Pilgrimage, 2nd edition, 1988.

[54] Douglas M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, (10):1–12, 2004.

[55] Huan Liu and Hiroshi Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[56] Susan W. Brenner. Organized cybercrime? how cyberspace may affect the structure of criminal relationships. *North Carolina Journal of Law & Technology*, 4(1), Fall 2002.