The Application of Formal Methods to Root Cause Analysis 0f Digital Incidents

Peter Stephenson, CISSP, CISM, FICAF
CeRNS – The Center for Regional and National Security
Eastern Michigan University

Abstract

Numerous current regulations and standards mandate incident response for virtually all segments of the private sector. According to most incident response experts there is the need to perform a root cause analysis (or "incident post mortem") following recovery from such incidents. To date there has not been a structured, formal approach to conducting this type of post incident analysis.

This paper proposes a methodology based upon formal modeling of the security processes in an enterprise under attack. The enterprise is segmented into manageable and security-relevant policy domains and the interactions of those domains including both pre- and post-incident states are modeled. The paper then shows how to analyze the nature of the state changes that occurred as a result of the incident and, finally, how to insert appropriate safeguards and countermeasures to prevent future occurrences of the same type of incident.

This methodology is based upon an ongoing research project, field testing, and other peer-reviewed papers. The formalism selected is Colored Petri Nets.

Introduction and General Approach

Organizations today are faced with two major issues regarding incident root cause analysis. First, standards, regulatory requirements, and best practices force most organizations to implement an incident response program. Best practices dictate that such a program includes some form of incident post mortem.

Second, there are few structured approaches for incident post mortems, and no such approaches in general use based upon formal modeling of the incident. Thus, credibility of the actual post mortem analysis is questionable until the next incident occurs and the lessons learned either result in effective controls or they don't.

Regulatory Requirements and Standards as Drivers for Post Incident Root Cause Analysis

Today there are numerous regulations and standards that mandate explicit response to information security incidents of most types. For example, the Basel Committee on Banking Supervision (the "Basel Committee") specifically mandates:

> A process for collecting and preserving forensic evidence to facilitate appropriate post-mortem reviews of any e-banking incidents as well as to assist in the prosecution of attackers. [BC03]

ISO 17799 in section 8-01-03-02 states:

> Incident Response procedures should manage the **1) analysis and identification action of the root cause**; 2) implementation of remedies to prevent recurrence; 3) collection of audit trails and similar evidence; 4) communication recovery team; and 5) reporting the incident to the appropriate authority. [emphasis added]

This is supported in the Sarbanes-Oxley Act. COSO (The Committee of Sponsoring Organizations of the Treadway Commission – "Internal Control – Integrated Framework") specifically supports this ISO 17799 provision with the following requirement:

> A security incident response process exists to support timely response and investigation of unauthorized activities.

Title V of the Gramm-Leach-Bliley Act of 1999 requires financial institutions to implement a comprehensive written information security program including eight elements that must be addressed in protecting customer information (emphasis added.

1. Access control
2. Physical security
3. Encryption of electronic customer information (especially in transit)
4. Change management procedures
5. Segregation of duties and employee background checks
6. Monitoring systems and procedures to detect security breaches
7. **Incident response program**
8. Disaster recovery

NIST (The National Institute for Standards and Technology) also makes incident post mortem an important portion of the incident response process [NI04]. The NIST-recommended incident life cycle is illustrated as:

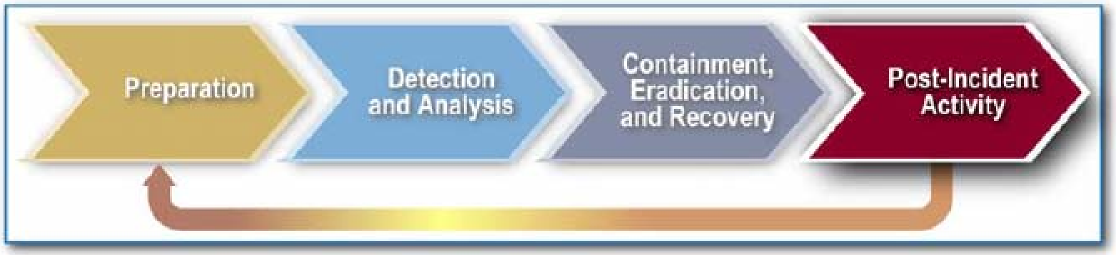


Figure 1. NIST Special Publication 600-81, Incident Response Life Cycle (Post-Incident Activity)

Clearly the notion of post incident root cause analysis is strongly embedded in regulatory requirements, standards, and recommended best practices, either directly or as part of a required incident response program.

The Need for a Formalized Approach to Post-Incident Root Cause Analysis

As most incident post mortems are informal affairs, the useful information gained may be equally informal and unpersuasive to management, regulators, and auditors. Most authorities on incident response and incident post mortem analysis agree that an important element of a post mortem is the root cause analysis. An equally important element is recommended safeguards and countermeasures. To those two elements we would add the need for verification that recommended countermeasures and safeguards have, indeed, been implemented.

Applying formal methods to the analysis of root cause is a logical next step in producing rigorous analyses. The following guidelines for a rigorous incident root cause analysis are proposed.

1. The analysis should be:
   - Comprehensive
   - Rigorous
   - Reproducible
   - Scalable
   - Relatively easy to perform
   - Reliable
   - Capable of producing clear results.
2. The analysis procedure should use readily available analysis and modeling tools.
3. The analysis procedure should be based upon proven and accepted mathematical precepts, methods and processes.
4. The results of the analysis should be acceptable for presentation to a court of enquiry at any level.

Problem Definition

The problem addressed is multidimensional. There is a need for a rigorous post incident root cause analysis procedure, including some guidelines for what such a procedure should produce. However, such a procedure presupposes a thorough understanding of such elements as security-relevant data flows throughout the affected enterprise, enterprise topology, security policies in place throughout the enterprise, existing (and, perhaps, failed) safeguards and countermeasures, the nature of the event, and the existence of acceptable evidence collected during earlier phases of the incident response life cycle. Each of those elements requires special attention.

Security-Relevant Data Flows

The problem of security-relevant data flows is closely allied with the topologies, policies and existing safeguards and countermeasures. These challenges are addressed through the use of security policy domains. A security policy domain is defined as:

Definition 1 – Security Policy Domain

A security policy domain consists of all of the elements of an enterprise that are subject to the same security policy.

Let $P =$ the set of all security policies on a bounded enterprise network

Let $E =$ the set of all elements on a bounded enterprise network

Let $p =$ a particular policy where $p \in P$

Let $e =$ a subset of elements where $e \subseteq E$

Let $P_{dom} =$ a security policy domain

$$P_{dom1} \subseteq E \mid e_1.p_1$$

Addressing these issues through policy domains, has important benefits:

- There is no need to map data flows on a device-by-device basis.
- Domains can be created with a single device if that device is worthy of its own domain based upon criticality and/or sensitivity.
- Whatever level of granularity is appropriate can be attached to the particular task to our model.

Once all of the policy domains have been identified, all of the inter-domain communications channels can be identified. All possible channels are of concern and the application of link analysis is appropriate to identify potential covert, or unauthorized, channels. However, simply because an inter-domain channel is possible, doesn’t mean that it actually exists. The application of countermeasures and safeguards can, effectively, block the channel preventing inter-domain data flows. It is, in fact, one of the modeling goals to identify unprotected covert channels and block data flows on them.

Existence of Acceptable Evidence

The existence of acceptable evidence in an incident post mortem is somewhat different from the existence of evidence during an incident. Typically, evidence that can help manage the incident is collected during the actual event. However, post-incident, evidence can be collected at a slightly more leisurely pace. This phase of the post-mortem is focused squarely upon identifying root cause and the investigation is not unlike the investigation of a computer-related crime. Since the results of the root cause analysis may be needed in a down-stream legal action, the post mortem investigation should be conducted under the same rules that would be followed if it were an investigation into a possible digital crime.

For the purposes of a post-incident investigation an accepted framework for the digital investigative process is necessary: the digital investigation framework developed by the Digital Forensics Research Workshop (DFRWS). This framework is illustrated in Figure 2 and is described in detail in [PS03] and [DFR01].

IDENTIFICATION	PRESERVATION	COLLECTION	EXAMINATION	ANALYSIS	PRESENTATION
Event/Crime Detection	**Case Management**	**Preservation**	**Preservation**	**Preservation**	Documentation
Resolve Signature	Imaging Technologies	**Approved Methods**	**Traceability**	**Traceability**	Expert Testimony
Profile Detection	**Chain of Custody**	Approved Software	Validation Techniques	Statistical	Clarification
Anomalous Detection	**Time Synch.**	Approved Hardware	Filtering Techniques	Protocols	Mission Impact Statement
Complaints		**Legal Authority**	Pattern Matching	Data Mining	Recommended Countermeasure
System Monitoring		Lossless Compression	Hidden Data Discovery	Timeline	Statistical Interpretation
Audit Analysis		Sampling	Hidden Data Extraction	**Link**	
		Data Reduction		Spatial	
		Recovery Techniques			

Figure 2. The DFRWS Digital Investigative Framework

The columns are referred to as "Classes" and the individual cells as "Elements." The elements in bold typeface are consider to be required as the general case in a digital investigation, while the remaining elements apply as necessary based upon the environment of the investigation.

The classes are not necessarily performed in order, since some classes are performed repeatedly as new evidence surfaces during the investigation. However, all classes must be represented in the investigative process. The purpose of the Framework is to add structure to the digital investigative process and to help ensure that all digital evidence is gathered and managed properly. For each of the cells there are acceptable processes that have found applicability in digital investigations and have been upheld in courts of enquiry. The DFRWS Framework is not a process. Rather, it is a loose taxonomy for digital investigation processes.

Supporting the DFRWS Framework is the End-to-End Digital Investigation (EEDI) process [PS03]. The End-to-End Digital Investigation process is a collection of generalized steps to be taken in conjunction with the DFRWS Framework. While the Framework gives a roadmap for addressing those issues comprising a structured investigation, the EEDI process provides a set of steps the investigator must perform in order to collect and analyze digital evidence.

The End-to-End process details consist of:

- Collecting evidence
- Analysis of individual events
- Preliminary correlation
- Event normalizing
- Event deconfliction
- Second level correlation (consider both normalized and non-normalized events)
- Timeline analysis
- Chain of evidence construction
- Corroboration (consider only non-normalized events)

It should be noted that there are other investigative frameworks [CS03] that may be equally applicable for collection and management of electronic evidence in the context of an incident post mortem. The EEDI process is emphasized because it focuses upon the analysis of events. EEDI tends to address the Collection, Examination and Analysis classes of the DFRWS Framework. One might consider EEDI to be a lower level of abstraction in an overall investigation. For example, it would be appropriate to include most of the steps outlined by Carrier and Spafford in the first phase of EEDI: Collecting evidence.

Approach

The problem of post-incident root cause analysis is a superset of the EEDI process. Once the investigation has been complete, the process of modeling and analyzing the processes that comprised the incident must begin. It is important here to note that an incident is a process. It is the process of performing the steps that allow the perpetrator of the incident to conclude a series of events that leads to some undesired conclusion. These steps could be performed automatically, as in a virus or worm attack, or they could be performed manually by an attacker. By collecting and analyzing the evidence and creating models of this process consistent with the evidence, the pre- and post-incident states of the victim system, as well as the processes that allowed the state change to occur, can be simulated.

Defining a Security Incident

The definition of a computer security-related incident is one that has eluded investigators and incident response specialists. In this paper, a computer security incident will be formally defined as:

Definition 2 – Computer Security Incident.

A computer security incident is a change of state in a bounded computer system from the desired state to an undesired state, where the state change is caused by the application of a stimulus external to the system.

Let A = the set of all possible operating states of a bounded computer system S

Let $\alpha_{cor} =$ the desired operating state of S where $\alpha_{cor} \in \mathrm{A}$

Let $\alpha_{pre} =$ the pre-incident operating state of S where $\alpha_{pre} \in \mathrm{A}$

Let $\alpha_{pre} = \alpha_{cor}$

Let $\alpha_{post} =$ the post-incident operating state of S where $\alpha_{post} \in \mathrm{A}$

Let $\beta =$ the set of external stimuli applied to S

Let $I =$ a computer security incident

$$I = \alpha_{post} \big| \exists \alpha_{pre}.\beta \Rightarrow \alpha_{post} \neq \alpha_{cor}$$

Notice that the definition makes no judgment as to the number or nature of the external stimuli, nor does it address their source(s) or nature(s). It simply is sufficient that some external (to the bounded system *S*) force caused a state change from desired to undesired in *S*. Thus, the event could be a hacking incident, a virus or worm, or a case fraud where the undesired state change comprises theft, compromise, deletion, alteration or other abuse of computer data. The perpetrator could be internal of external to the organization.

Computer security incidents are generally thought of as requiring immediate response, and, from the perspective of the incident response team, that is correct. From the perspective of the post-incident analysis, though, it makes no difference whether immediate response was required or not. It is sufficient, simply, that an undesired state was induced in the victim system through external means. This could, of course, include non-security incidents caused by such problems as misconfigured routers or other internetworking devices.

However, since, at the time of the post-incident investigation the root cause, by definition, is unknown, the discovery of a non-malicious act as the trigger for the event fulfills the mission of the investigation. Thus, malicious intent is not a requirement for a security incident.

One final note regarding bounded computer system *S*: this is not necessarily a full enterprise network. It may be a single victim device on the network. The stimulus is literally *external* to system *S.* Thus the attacker could reside within and launch the attack from within the enterprise. The attack still is *external* to the victim because it was spawned by the attacker and not by the victim system itself.

Identifying Inter-domain Communications Channels

The process of identifying inter-domain communications channels begins by identifying the security policy domains within the enterprise and external to it (where external domains may have impact on the victim). These identifications are based upon two criteria:

- Data classification, if present, and;
- Appropriate security characterization of the enterprise.

In most cases, the enterprise is “security flat.” In other words, there is no filtering used to separate domains based upon the nature of the data in them. There will usually be an external firewall separating the public domain (Internet) from the internal network, and, if there is a complex perimeter/DMZ network, there may be a second firewall as well. However, it is unusual for such inter-domain devices to be present on the internal network, even where such external domains as wireless networks and extranets touch the internal enterprise.

The source for domain identification is a set of interviews with data owners/custodians and systems administrators as well as network maps to understand how data flows and is used on the enterprise. A network mapping tool can then be used to “see” the network and help discover covert domains. This process may uncover external domains and domain connections about which the organization is unaware.

Once the security policy domains (either actual domains based upon actual configuration of the enterprise, or imputed domains based upon identified data flows, locations and uses) have been identified, shallow link analysis[1] is used to identify all of the inter-domain communications channels and extrapolate those known channels to potential covert channels. A typical inter-domain shallow link analysis is shown in Figure 3.

[1] “OntologyStream Inc Briefing & Elementary Tutorial on categoricalAbstraction (cA)”, Prueitt, Paul, Tool and tutorial on shallow link analysis can be obtained from: http://www.ontologystream.com/cA/elementaryTutorial.htm

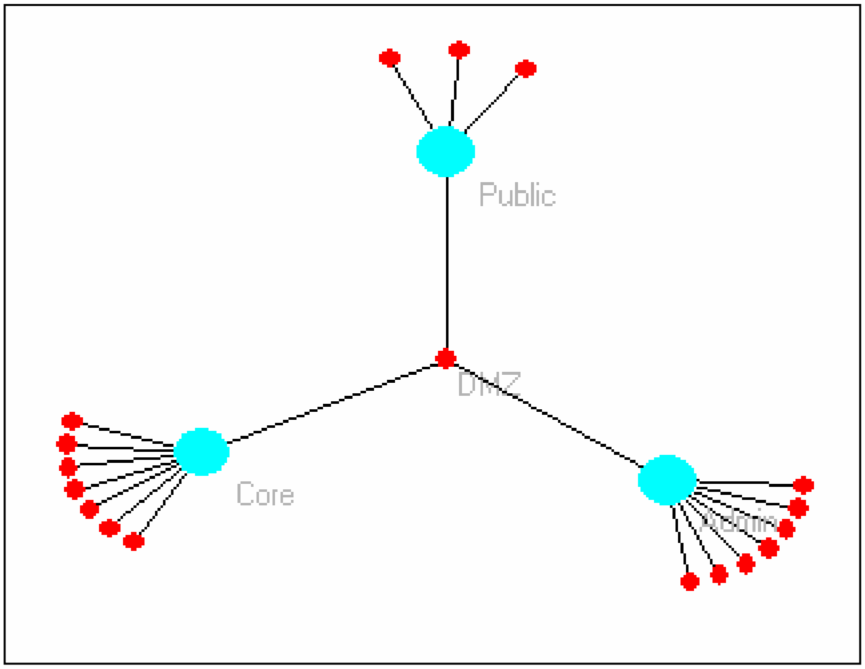


Figure 3. Shallow Link Analysis Showing Covert Channels Through a DMZ

Figure 3 shows the shallow link analysis of the channels between the Public domain, the Core domain, the Admin domain and the DMZ domain. Note that there are potential paths between the Public domain and the Core and Admin domains. These paths could be realized by compromising the DMZ. Additionally, the smaller dots on the Core, Public, and Admin domains represent other domains with direct connections to Core, Admin, and Public. Clicking on these dots in the analysis tool will reveal additional links.

The tool's data store was populated using domain and inter-domain channel information derived from interviews and network maps. The tool then analyzes all possible combinations of inter-domain links and maps the result. For this example twelve separate domains with known inter-domain communications channels were used.

Constructing Models and Performing Simulations

The modeling formalism used is Colored Petri Nets (CPNets). The existence of a formal definition for CPNets is required as a basis for simulation, but it is not necessary for the user to know or understand the formal definition in order to do useful work with CPNets.

Additionally, CPNets are graphical representations of formal mathematics. The user need not be conversant in formal math to use the CPNet tools. Since the nets are graphical representations, they are ideally suited to demonstrating complex processes to lay audiences. For this research CPNTools, a free tool set from the University of Aarhus in Denmark, the home of CPNets, was used.

Colored Petri Nets consist of places, transitions, tokens and arcs. Places describe the state of the system as with traditional Nets. Transitions, also as in traditional Petri Nets, describe the actions or changes in state of the system. Tokens are place markers that help us understand state changes. Arcs connect places and transitions, and identify how actions modify the state and when these actions occur. Because of the underlying mathematics in CPNets, a variety of data types can be applied to the Nets.

For illustrative purposes, security policy domains are represented as places and inter-domain communications channels as transitions and arcs. Where appropriate, constraints are placed upon the channels in the form of guards (Boolean statements) on the transitions. The appropriate conditions for state changes are described by means of variables on the arcs. A CPNet showing analysis of a worm infection is shown in Figure 4.

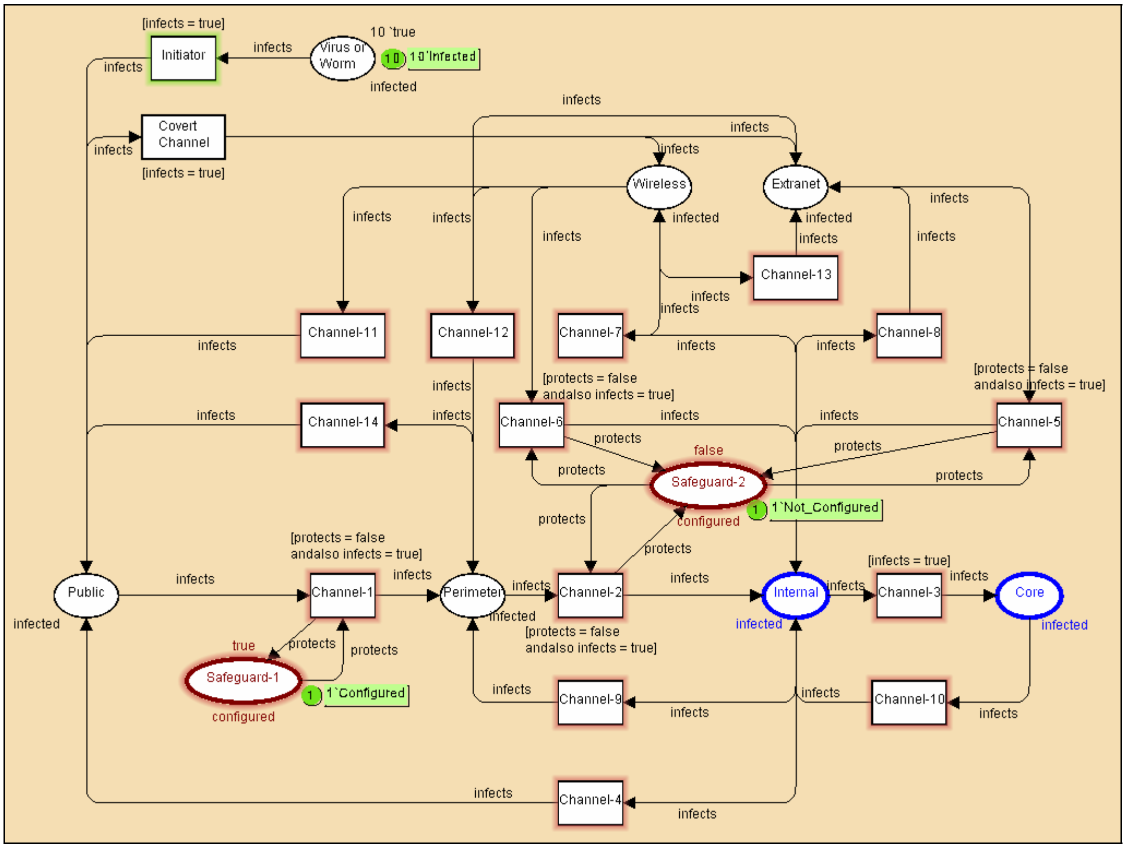
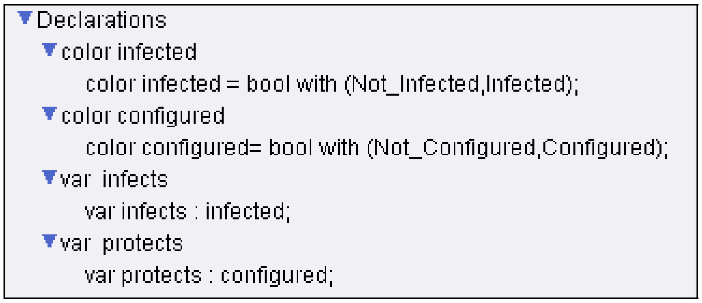


Figure 4. Simple Pre-Incident CPNet of a Virus or Worm Infection

In Figure 5 we show the declarations for this simple Net.



```
▼Declarations
  ▼color infected
      color infected = bool with (Not_Infected,Infected);
  ▼color configured
      color configured= bool with (Not_Configured,Configured);
  ▼var  infects
      var infects : infected;
  ▼var  protects
      var protects : configured;
```

Figure 5. Declarations for the CPNet in Figure 4

The CPNet in Figure 4 represents a worm infection in a mid-sized enterprise. It shows six security domains:

- Public – the public Internet
- Perimeter – the DMZ
- Internal – most of the internal network devices
- Core – critical and sensitive devices and the high-speed core routers and switches
- Wireless – the enterprise's wireless network
- Extranet – connection to external partners

Note that in Figure 4, Safeguard-2, an internal inter-domain filter applied to several paths that the virus or worm might take, is shown in the state, "Not Configured." This lack of inter-domain protection allowed the worm to spread throughout the enterprise, even though there was a configured perimeter firewall (Safeguard-1). Because the internal safeguard(s) were not configured (indeed, they were not present), the worm was able to enter the enterprise through the wireless network and cause an undesired state change in the many devices within the enterprise.

The inter-domain communications channels are represented in Table 1. Note that these channels include covert and back channels as discovered through shallow link analysis.

SOURCE DOMAIN	DESTINATION DOMAIN	CHANNEL	COMMENTS
Public	Perimeter	1	
Perimeter	Internal	2	
Internal	Core	3	
Internal	Public	4	This is a feedback loop - infected devices internally can spread the infection back into the public Internet
Extranet	Internal	5	
Wireless	Internal	6	
Internal	Wireless	7	This is a feedback loop - infected devices internally can spread the infection back into the public Internet
Internal	Extranet	8	This is a feedback loop - infected devices internally can spread the infection back into the public Internet
Internal	Perimeter	9	This is a feedback loop - infected devices internally can spread the infection back into the public Internet
Core	Internal	10	Feedback loop between two internal security domains - results in re-infection
Wireless	Public	11	This is a feedback loop - infected devices internally can spread the infection back into the public Internet
Wireless	Perimeter	12	Feedback loop between two internal security domains - results in re-infection
Extranet	Perimeter	12	Feedback loop between two internal security domains - results in re-infection
Wireless	Extranet	13	Feedback loop between two internal security domains - results in re-infection
Perimeter	Public	14	This is a feedback loop - infected devices internally can spread the infection back into the public Internet
Virus or Worm	Public	Initiator	Initial infection vector
Public	Wireless	Covert Channel	Back Channel that introduces the infection to the enterprise
Public	Extranet	Covert Channel	Back Channel that introduces the infection to the enterprise

Table 1. Inter-Domain Communications Channels for the CPNet in Figure 4

Figure 6 illustrates the post-event state of the enterprise.

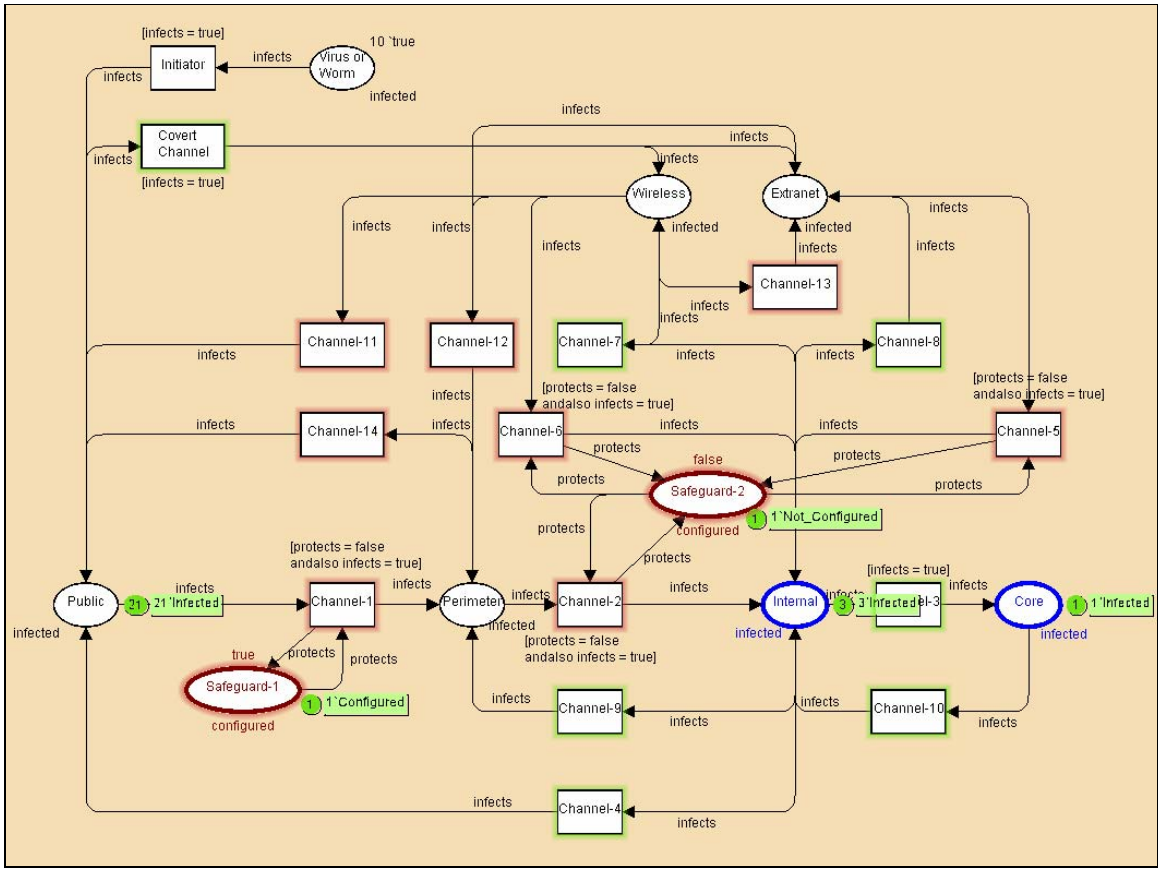


Figure 6. **Simple Post-Incident CPNet of a Virus or Worm Infection**

Note that the state of the Public domain has changed to Infected, with a total of 21 tokens in that place. Each token represents a state change in this Net and may be interpreted to represent re-infection by the virus or worm, as would be expected on the Public Internet.

However, both the Internal and Core domains have experienced a state change which may be interpreted as infection in this Net. Thus, the model, after simulation, reveals that the safeguards were not effective.

This Net represents the state of the enterprise after many cycles of simulation. However, the cycle can be simulatd a single step at a time, showing clearly where the worm or virus entered the enterprise based upon the step-by-step state changes of the policy domains represented in the model. Additional detail can be added to the model to represent timing issues between domains, functions that inhibit or accelerate inter-domain data flows, etc.

Final Steps

After collecting and analyzing evidence, describing security policy domains and inter-domain data flows, performing shallow link analysis, and modeling and simulating the results, the next step is to insert appropriate safeguards into the model and re-simulate. Thus, all of the requirements of the root cause analysis have been met, resulting in an understanding of how the incident occurred, a formal proof of the analysis, which is in a form that can be presented easily to a lay audience, and the ability to recommend and test countermeasures and safeguards.

Future Work

It remains to develop these techniques into an easy to use, integrated tool set. Additionally, there are numerous possibilities for extending the accuracy of the modeling technique to identify entry points more accurately based upon application of Bayesian probability functions embedded within the CPNet.

Although the theory behind these techniques has been field-tested successfully, there are several opportunities for extending the modeling further to include predictive elements that would allow analysts to proactively identify likely successful attacks and implement safeguards before the fact.

About the Author

Peter Stephenson is the director of information assurance at CeRNS – the Center for Regional and National Security at Eastern Michigan University. He is the author of several books and numerous articles in computer and information security publications. He may be reached at peter.stephenson@emich.edu.

References

[BC03] Basel Committee on Banking Supervision, “Risk Management Principles for Electronic Banking.” July, 2003., retrieved 27 June 2004, http://www.bis.org/publ/bcbs98.pdf.

[CS03] Carrier, Brian, and Eugene H. Spafford (2003). "Getting Physical with the Digital Investigation Process." *International Journal of Digital Evidence*. Volume 2, Issue 2 (2003): retrieved 22 November 2003, http://www.ijde.org/current_home.html.
.

[DFR01] Report from the First Digital Forensic Research Workshop. *DTR-T001-01 FINAL A Road Map for Digital Forensic Research*. Final version, November 6, 2001. http://www.dfrws.org.

[NI04] Grance, Tim, Karen Kent, Brian Kim. ”NIST Special Publication 800-61, Computer [NI04] Security Incident Handling Guide.” National Institute of Standards and Technology. January, 2004.

[PS03] Stephenson, Peter (2003). Modeling of Post-Incident Root Cause Analysis. *International Journal of Digital Evidence.* Volume 2, Issue 2, http://www.ijde.org.