Honeytraps, A Network Forensic Tool

Alec Yasinsac and Yanet Manzano
yasinsac@cs.fsu.edu manzano@cs.fsu.edu

Department of Computer Science
Florida State University
Tallahassee, FL 32306-4530
850.644.6407 (voice), 850.644.0058 (fax)

Abstract

Two new fields within Communication Systems and Networks, born in the practioner realm, have quietly grown in importance in the Internet age: Computer Forensics and Deception technology. These two fields share the goal of collecting information about computer mischief, but with dramatically contrasting desired outcomes. To date, no published results document efforts to leverage the similarity in these areas. In this paper, we detail a method to utilize deception technology to enhance computer and network forensic capabilities.

Keywords: Communication Systems, Networks, Computer Forensics, Deception Technology, Information Security

1. Introduction

There are two primary reasons for gathering information about computer crimes. First, information is gathered to allow criminals to be prosecuted in court. This is the goal of Forensics. Second, information is gathered that will help create counter-measures to prevent crimes. This is the goal of deception technology such as Honeypots. In this paper, we address some specific ways these two information-gathering technologies can be combined.

1.1. Background

For years, computer and network security experts (whitehats) have fought to stay ahead of computer criminals (blackhats). As blackhats became more skilled and computers became more powerful, conventional security measures became less effective. This perpetual action-response-reaction cycle evolved into a new field of study known as Computer and Network Forensics. (CNF). CNF is the art of discovery and retrieval of information about computer related crime in such a way that the gathered information is admissible in court.

There are two sides to CNF efforts. The first is to assess the impact of the malicious or suspect act or acts. In order to bring a computer criminal to justice, it must be possible to show that sufficient damage has been done so that the act can be accurately classified as a crime. Often, there is an economic threshold associated with statutes that govern computer crime.

The second part is to gather information that legally binds the act or acts that caused the damage to the perpetrator. This is the better known aspect of computer crime investigation; the standard "Who dun' it" component.

In response to innovative computer criminals, CNF techniques have become highly sophisticated and CNF tools are increasingly effective. In addition to putting computer criminals in jail, CNF techniques have enabled whitehats to learn valuable information about blackhats' techniques and methods and to formulate protection and defense mechanisms, tools, and techniques.

Until recently, the relationship between CNF and mainstream computer and network security techniques has been vague at best. By their nature, security efforts traditionally depend on actions that are taken before an attack to protect resources or information from malicious access or use. This is done through access control techniques, encryption, and vulnerability assessment mechanisms. More recently, significant effort has been focused on providing attack detection and response technology that works during suspected attacks to protect resources [VK99, JOU00].

Alternatively, CNF traditionally has had a different focus from both of these two perspectives. First, CNF is concerned with gathering information about attacks and perpetrators rather than directly protecting resources or information. Consequently, the second fundamental difference is that CNF has historically dedicated its efforts to actions taken after-the-fact, i.e. after malicious or suspicious activity has occurred, rather than activity that occurs before or during attacks.

An early effort to systematically connect CNF and security was given in [MY01]. That work fundamentally extended the scope of CNF by proposing policies and techniques that could be implemented before an attack occurred that facilitate the CNF effort both during and after malicious or suspicious activity occurred. This paper is an extension of that idea based on using independently implemented systems to gather information that enable CNF experts to put computer criminals that attack production systems into jail.

Deception technology is well known in the security practitioner arena [DTHP] and a mathematical foundation for its effectiveness was prescribed in [COH01]. The related concepts of deception security, Honeypots, and Honeynets [HN] have been the subject of organized investigation for several years. We coin the term "honeytraps" to reflect the tools that fall into any of these categories. Honeytraps allow us collect information about blackhat activities without putting a real system at risk. In this paper we show how honeytrap technology can be a valuable elements of a forensic toolkit.

2. Honeytraps

Honeytraps are systems (Honeypots or Honeynets) that are designed to be compromised. Honeypots are host systems that attract[1] intruders to enter the host by emulating a known vulnerability. Essentially, they are modified production systems that create contained environments where intruder actions can be more safely monitored and documented Their have no real, valuable data or information. Their main goal is to capture and analyze data in order to learn about the blackhat community.

[1] We employ a soft interpretation of "attraction" here. The notion of attracting intruders is counter-productive in most senses.

Honeynets, on the other hand, are a network of interconnected production and honeypot nodes. They protect the production resource only by distracting the intruder from the real target. Like Honeypots, Honeynets are designed to collect information from intruders and attempted intruders while containing the operations that these intruders can perform.

2.1. Pitfalls of Honeytraps

There are several potential pitfalls of honeytraps, one based on its foundational goal of being a system that is established to be compromised. A concern is that once an intruder enters the honeytrap, they may be able to utilize some component of the honeytrap for an illicit purpose, e.g. as a zombie in a distributed denial of service attack. *Containment* involves the policies, architecture, procedures, and techniques taken by a honeytrap creator to protect against such an occurrence.

A second concern is that once the blackhat enters the honeytrap, they may be able to attack the honeytrap itself, shielding their actions from the designed honeytrap monitors or by destroying or modifying the honeytrap activity logs. There are many discussions of how to avoid these and other honeytrap pitfalls in the literature and on relevant web pages. For the purposes of this paper, we posit without proof, that these pitfalls can be effectively overcome.

2.2. Data Capture.

Once a blackhat penetrates a honeytrap, there must be mechanisms in place to detect and record the actions that the intruder takes. Detecting and recording that activity is termed *Data Capture*. Data Capture should record every possible aspect of the blackhat activity, from keystrokes to transmitted packets. The purpose of data capture is to collect data to determine tools, tactics, habits, and motives of specific blackhats, and of the blackhat community [KEH].

2.3. Honeytrap Uses

Honeytrap are intended to let the blackhats in and allow them to operate in order to monitor their actions. As information is collected, blackhats are profiled and their techniques are analyzed.

2.3.1 Documenting Blackhat Techniques

For years whitehats have dedicated efforts and resources to study the blackhat community with the ultimate goal of learning blackhat techniques. From earlier efforts, such as hacker surveys [Me98], to the cutting edge technology such as that implemented in the honeynet project, whitehats have used their knowledge to investigate the mystery of blackhats and the working of the hacking process. In 1999, MacClure [HE99] shed some light on this subject when he documented the process of hacking by breaking it down into stages that most blackhats goes through during an attack. The anatomy described by MacClure includes four stages: proving, invading, mischief, and covering tracks. Documenting the blackhats activities during these four stages allows us to create a signature that can be used to identify a specific blackhat.

2.3.2 Profiling Specific Blackhats

Through the literature produced from previous research, blackhats techniques, tactics, motives and psychology have been documented [KEM00]. We now use this information to create signatures to characterize specific blackhats. For example, suppose our blackhat is a script kiddy. Script kiddies are inexperienced blackhats that try to break into systems using scripts created by knowledgeable blackhats. A signature for this blackhat may include, for example, level of skill, methodology, tactics, tools, and other information such as the originating site for scripts.

2.4. Letting Them In or Inviting Them In

An essential element of deception technology is that hackers must enter the trap in order to for the trap to gather information. By many reports, hacking and probing is sufficiently widespread that simply placing a computer on the Internet will naturally result in intruders entering the computer. Still, there is no guarantee that there will be enough of any interesting types of hacking in the computer to allow effective information gathering.

In order to be effective, honeytraps may need to *generate* hacking traffic by attracting intruders into the honeytrap. As we alluded to earlier, attracting blackhats into a honeytrap is not without risks and honeytrap operators may be liable for damage to other systems if the blackhats are able to turn the honeytrap into an attack engine.

From a forensics standpoint, attracting intruders raises legal issues as well. If the attraction is sufficiently overt, the intruder's entry may be justified; in other words, honeytraps may be legally considered fair game to intruders. Additionally, even if the intrusion is found to be illegal, the intruder may be able to claim that the attraction of the honeytrap served as entrapment. A challenge of combining forensics and honeytrap technologies is to resolve these issues.

2.5. Honeytrap Approaches

Honeytraps are inherently flexible and can be implemented in a wide variety of ways. There are three primary considerations that guide the approach that we recommend: 1) System vulnerabilities, 2) Operating system attacks and 3) Network system attacks.

For the first, we can configure honeytraps to identify a specific vulnerability or set of vulnerabilities within a host or system. For example, the honeytrap "BackOfficer Friendly" [BOF] can be configured to emulate the specific vulnerability known as Back Orifice. When blackhats find this honeytrap and recognize that the Back Orifice vulnerability is open, BackOfficer Friendly carefully tracks their activities in the honeytrap to see how they exploit the vulnerability. The tool extends the process by assessing the impacts of the recorded actions.

In the second category, we can configure a honeytrap to mimic an operating system to determine what vulnerabilities these systems have. The honeytraps Mantrap [MT] and CyberCop [CCS] are examples of automated tools that implement honeypots that assess operating system vulnerabilities. Each of these tools can concurrently simulate multiple operating systems vulnerabilities.

In the final category, we can configure a honeytrap to simulate a network system and monitor the system behaves and how its components interact under attack. Honeytraps are an effective technique to test changes in systems, such as addition of new software or significant configuration changes to determine possible risk and vulnerabilities before they are implemented in the real system.

3. Honeytraps Used For Forensics?

With the introduction of honeytraps, the face of information gathering changed, putting whitehats in the

offensive rather than the defensive mode. The purpose of honeytraps is to gather intelligence about the enemy to learn the tools, tactics, and motives of the blackhat community [HN]. To date, the information collected in honeytraps has not been intended for presentation in court. In order to use the information collected in honeytraps to prosecute the blackhat there are numerous legal issues to deal with.

As we discussed earlier, when an intruder is attracted (no matter how subtle that attraction may be) into a honeytrap, the honeytrap owner assumes liability for the actions the intruder takes on the honeytrap. For example, if the intruder is able to turn the honeytrap into a zombie to affect a distributed denial of service attack, the subject of that attack may claim damages from the honeytrap owner. Containment technology employing wrappers and sandbox techniques reduce the vulnerability, but are far from perfect.

Secondly, if an intruder is attracted into a honeytrap, it is unlikely that the intrusion itself can be prosecuted as a crime, even if the activity that the intruder engages after entry is clearly malicious. Proving crimes relative to invited participants is more legally complicated than for acts carried out by uninvited intruders.

Additionally, honeytraps are not real systems, they contain no valuable data, and they have no real users. As a result, there is no real economic impact and no real damage that can result from honeytrap intrusions. Honeytraps are created to be attacked, so it is unlikely that an intruder could be prosecuted for activities they undertake within a honeytrap since it would be difficult to categorize the results of their activities as a crime.

Even if a crime can be established, if the intruder was attracted into the honeytrap, it is a good chance that they will be

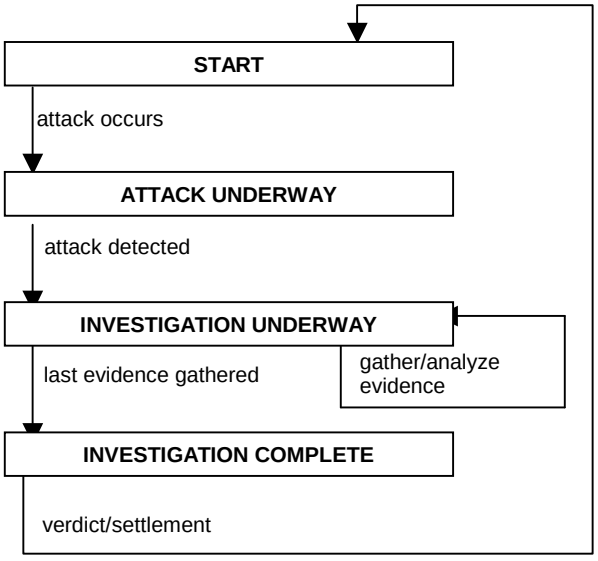

Figure 1

able to employ an entrapment defense. Even subtle attractions can be used to defeat prosecution via anti-entrapment laws.

Finally, honeytrap operators must deal with legal issues related to privacy. While privacy issues are not well defined on the Internet (or in society in general) honeytrap operators may face invasion of privacy claims either in response to their attempts to prosecute intruders, or independently from malicious or non-malicious intruders that do not desire to have their activities or identities revealed.

4. Computer and Network Forensics

4.1. Overview.

Computer and Network Forensics, is the art of retrieving information about a crime in such a way as to make that information admissible as evidence in court. The CNF ultimate goal is to provide sufficient evidence to allow the blackhat to be successfully prosecuted. CNF techniques are used to discover evidence in a variety of crimes ranging from theft of trade secrets, to protection of intellectual property, to general misuse of computers.

4.2. Forensics State of the Art

As we noted earlier, previous CNF efforts included only after-the-fact activities as modeled in Figure 1. In the traditional model, gathering potential evidence started only after an attack was detected and an investigation was set in motion. In [MY01] a variation was introduced to this cycle that allows the forensic process to be continuous. Implementation of these policies provides a new forensic model shown in Figure 2.

4.3. Honeytraps as Forensic Tools

In the previous section we introduced the traditional forensic model and then described how this model changed with the additions introduced in MY01. In the next sections we introduce two architectures that allow honeytraps to be used as CNF tools. We show how these developments transform the forensic model to create the parallel and the serial forensic architectures. In addition, we discuss the role of honeytraps in the forensic process.

As we have noted, honeytraps are designed to provide insight into blackhat methods, tactics, and targets. To gather this

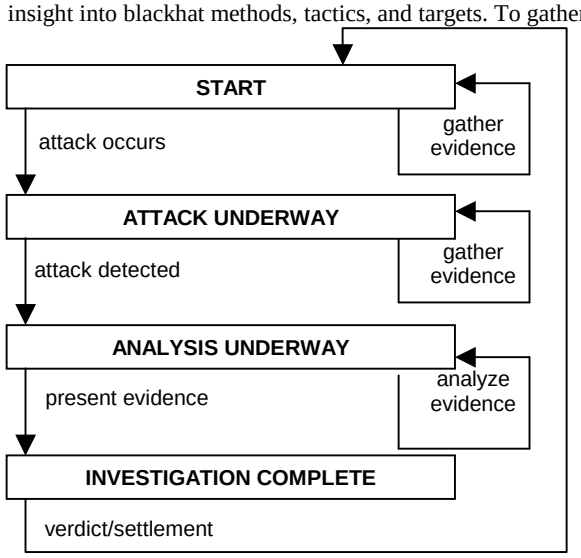

Figure 2. A New Forensic Model

information, blackhats must be tracked through the system, with every action being recorded for later analysis. This information provides a basis to determine how the blackhat works and allows the analyst to predict what this blackhat, or a class of blackhats, may do in the future.

We notice that information gathered from Honeytraps may be used to develop a profile of each blackhat that is monitored. By tracking the actions, a signature for attacks can be created that we can use to identify and prosecute the blackhat.

4.3.1 Architectures

Honeytraps come in many shapes and sizes. They are highly configurable and therefore can be designed to meet the needs and capabilities of a wide variety of specific systems. Once the Honeytrap is designed, the architecture of how to connect the Honeytrap to the Internet in reference to the production system must be determined. Two architectures that facilitate the forensic investigation are the serial and parallel architectures.

4.3.1.1. Serial Architecture

The serial honeytrap architecture works by placing the honeytrap between the Internet and the production system as shown in Figure 3. In this configuration, the honeytrap acts as a firewall. All recognized users are filtered to the production system while blackhats are contained in the honeytrap. The blackhats' activities are monitored and all the information collected is routed to another system that is protected by a firewall, to ensure the integrity of the data.

The serial architecture forces the blackhat to go through the honeytrap to attack the production system thus exposing all attackers to the honeytrap monitoring techniques. This may also enhance tracing capability, since it may be possible to follow blackhats as they transition between the honeytrap and the production system, making it easier for the forensic investigation to match the blackhat in the honeytrap to the blackhat in the production system.

There are numerous detractors to the serial architecture. We first notice that it is resource intensive. One of the important characteristics of Honeytraps is that they need not deal with real users, thus reducing the volume and complexity of monitoring. However, in the serial connection the honeytrap must handle all

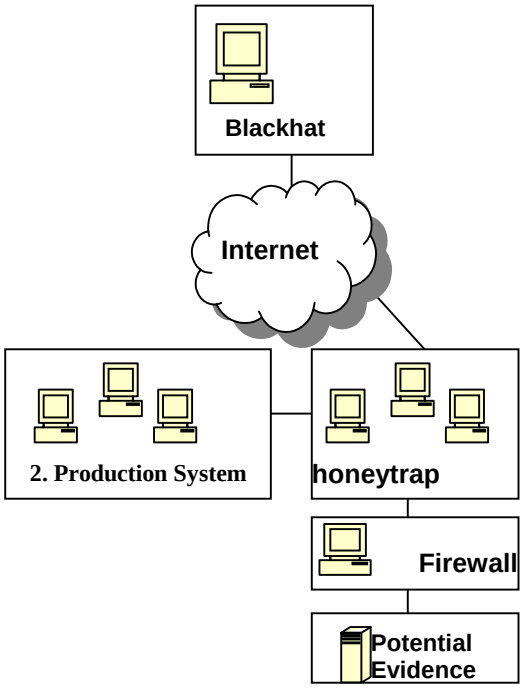

Figure 3: Serial Architecture

traffic going into the production system and reroute the authorized user to the production system. Additionally, were it easy to contain intruders in a firewall, we would not need honeytraps. This architecture runs the fundamental risk that intruders that it attracts into the honeytrap, may subsequently successfully attack the production system in in spite of the best containment efforts of the honeytrap.

4.3.1.2. Parallel Architecture

Alternatively, the parallel configuration allows the honeytrap to be independent of the production system as shown in Figure 4. As with the serial configuration, the information gathered about blackhat activities in the honeytrap is rerouted to a separate, protected system.

This architecture is less resource intensive so it can be implemented in a system with fewer resources. As with the

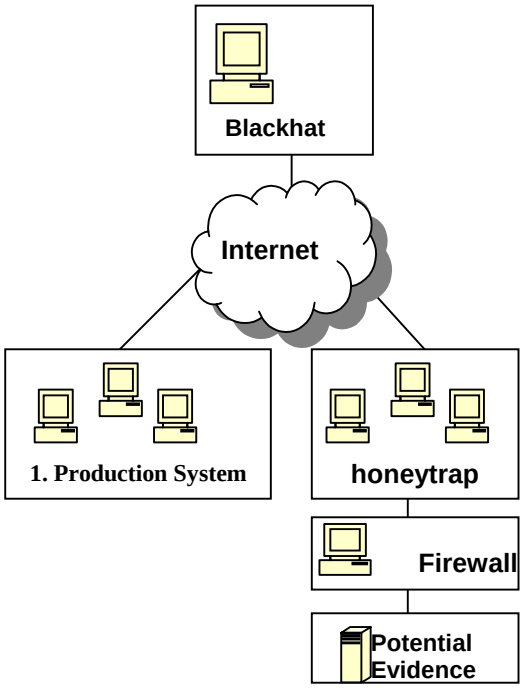

Figure 4: Parallel Architecture

serial architecture, here are several drawbacks with the parallel honeytrap architecture. The first is that for the honeytrap to be useful during the forensic process, both systems (honeytrap and production) must have been attacked independently. Configuring the honeytrap so that it is likely that an intruder would enter or probe the honeytrap before or shortly after entering the production systems is tricky, and again leads us into possible entrapment scenarios.

Secondly, under the parallel honeytrap architecture it is likely to be more difficult to connect an intruder in the honeytrap to the intruder in the production system if the honeytrap is implemented in the parallel configuration, since there is no direct connection between them as we had in the serial architecture.

4.3.2 Forensic Models for Serial and Parallel Honeytrap Architectures

The introduction of honeytraps as forensic tools and the proposition of the new architectures to facilitate the forensic process provides additional adjustments to the forensics process model for the serial and parallel forensic architectures.

In the Serial Forensic Model given in Figure 5, the forensic process begins when the blackhat has entered the honeytrap. Once the blackhat has gained access to the honeytrap, the Forensic Alert System (FAS) is activated. The FAS is a system

integrated into the honeytrap whose main purpose is to alert the

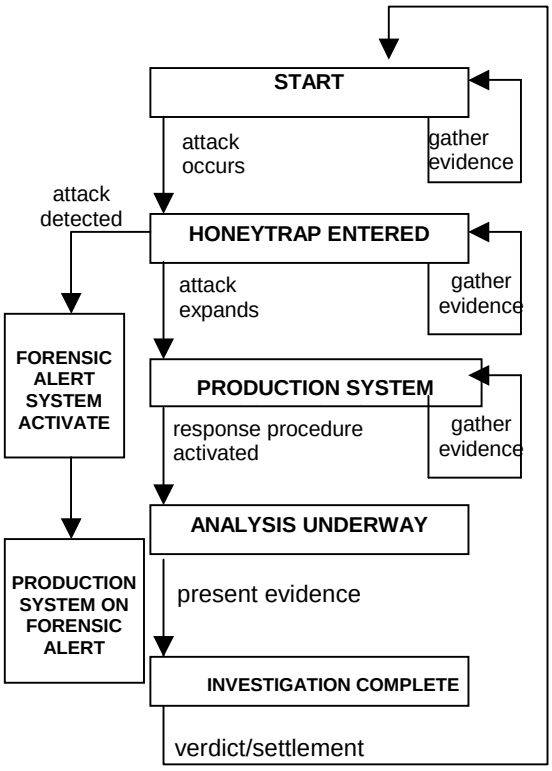

Figure 5: Serial Forensic Model

production system of the attack in progress and to start the monitoring of the blackhats' activities. The production system is on forensic alert after it receives the alert from the FAS, which could help to better prepare in case the attack expands into the production system.

If the attack expands into the production system, activate the response procedure to handle the attack [MY01]. After the situation is contained, move on to the investigation. When sufficient evidence is collected in the honeytrap and in the production system, the analysis begins and a report is produced that includes all the evidence and findings enables prosecution of the blackhat.

The states of the parallel forensic model, shown in Figure 6, are very similar to those of the serial forensic model with one mayor difference; in the parallel forensic model there are two processes ongoing concurrently. Process A is the honeytrap forensic process (HTFP), and Model B is the production system forensic process (PSFP).

The HTFP begins once the blackhat has entered the honeytrap and the FAS is activated. An

alert is immediately sent to the production system, and the monitoring of the blackhat's activities begins. Once the blackhat's activities are detected, a forensic investigation is performed with the information collected, and the results are safely stored until needed.

The PSFP begins once the production system has been compromised and the attack is detected. If the honeytrap is attacked first, then the production system is already on forensic alert, making it easier for the attack to be detected. Once the intrusion response procedure is activated and the situation is contained, the forensic analysis begins with the evidence gathered on the production system and the information collected from the honeytrap, if available. A complete report from the analysis is generated that includes all the evidence and findings that can be used to take legal action against the blackhat.

4.4. The Forensic Investigation

In both architectures, the forensic investigation procedure and goal is the same. The forensic investigation is broken down into two separate investigations. The forensic investigation begins in the evidence collected from the honeytrap, which will refer to as Honeytrap Forensic Investigation (HTFI). The second investigation is based on the on the evidence collected from the production system, which we will refer to as Production System Forensic Investigation (PSFI). Both

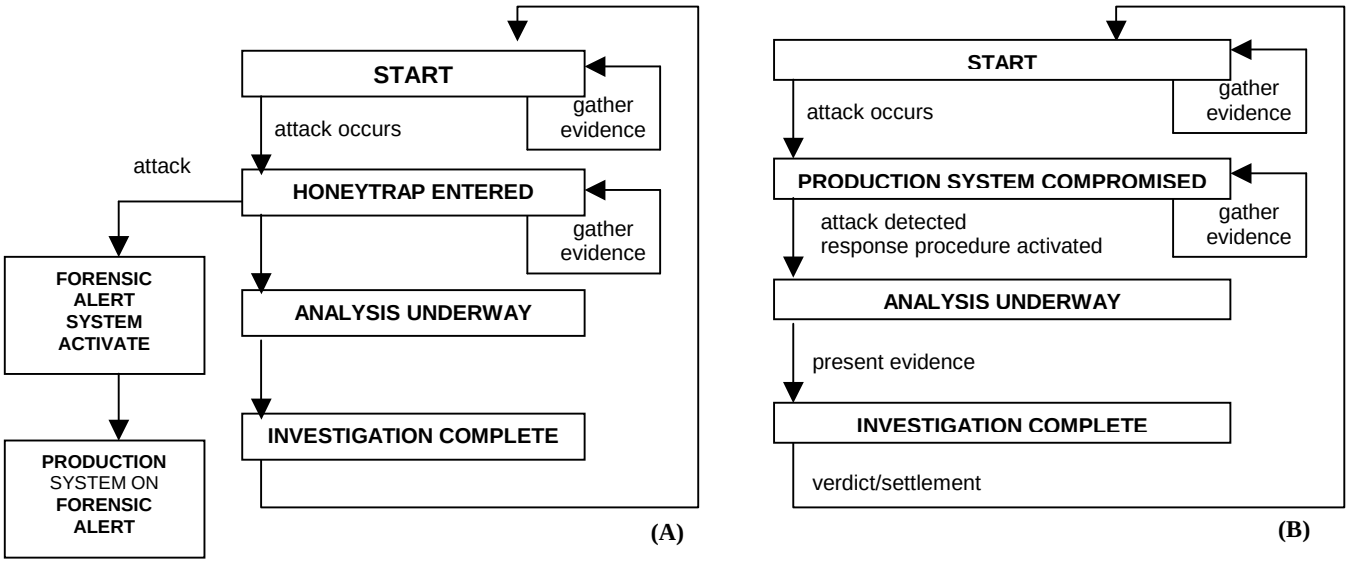

Figure 6: Parallel Forensic Model

investigations will produce a piece of the puzzle.

The goal in the HTFI is to produce a damage report and a signature for the blackhat. For example, suppose A is the blackhat that broke into the honeytrap, then the HTFI will produce:

1) A -> identity
2) A -> tactics
3) A -> tools
4) A -> targets
5) A -> other info

Because less information about the blackhat is available in the production system, the blackhat's signature may only be partial. For example, suppose B is the blackhat that broke into the production system, then the PSFI might include:

1) B -> tactics
2) B -> tools
3) B -> targets
4) B -> other info

An essential element of this investigation is to determine the identity of the intruder in the production system. The PSFI provides blackhat B's partial signature and a damage report, but not blackhat B's identity. The HTFI establishes blackhat A's identity, but A cannot be charged because he was in a honeytrap, so no real damage can be shown in court. So, we need the identity of blackhat B to charging him with the damage report.

The question is, "How can we use what we know about blackhat A to discover who blackhat B is?" The answer is that if we can show that the tactics, tools, targets, and other information signatures of intruder B are identical to those of intruder A, we may be able to make a compelling argument that A and B are the same person. If so, since the identity of intruder A is known, the match would enable the case to be pursued.

5. Conclusion

In this paper we present a new model for Computer and Network Forensics (CNF). We develop two architectures for utilizing deception technology (and coined the term Honeytraps to describe them) in CNF investigations. We give an implementation model for these architectures that illustrates how these two mutually exclusive technologies can be combined to improve Computer and Network Forensic Investigations.

6. Acknowledgements

Thanks to Fred Cohen for his insightful comments on a variety of topics as we prepared this paper.

7. Bibliography

[BOF] "Back Officer Friendly", http://www.nfr.net/products/bof/index.html

[CCS] "Cyber Cop Sting", http://www.nai.com

[COH01] Fred Cohen, "A Mathematical Structure of Simple Defensive Network Deceptions", http://all.net/journal/deception/mathdeception/mathdeception.html

[DTHP] "The Deception Toolkit Home Page", http://www.all.net/dtk/

[JOU00] Y. Jou, F. Gong, C. Sargor, X. Wu, S. Wu, H. Chang, and F. Wang, "Design and Implementation of a Scalable Intrusion Detection System for the Protection of Network Infrastructure", DARPA Information Survivability Conference and Exposition 2000, Jan 25-27, 2000, Vol. 2, pp 69-83

[HE99] Stuart MacClure, Joel Scambray, George Kurtz, "Hacking Exposed: Network Security Secrets and Solutions", Osborne/McGraw-Hill 1999

[KEH01] "Know Your Enemy: Honeynets", http://www.honeynetproject.org/papers/honeynet/

[HN] "Honeynet.org", http://www.honeyproject.org/

[Me98] Carolyn P. Meinel, "The Happy Hacker, A Guide to Mostly Harmless Computer Hacking", 2nd Ed. Scientific American, Inc. 1998

[MT] http://www.mantrap.com

[MY01] Yanet Manzano and Alec Yasinsac, "Policies to Enhance Computer and Network Forensics", 2nd Annual IEEE Systems, Man, Cybernetic Information Assurance Workshop, June 2001.

[VK99] Vigna and Kemmerer, "NetSTAT: A Network-based Intrusion Detection System" "Journal of Computer Security", Volume 7, Issue 1, 1999