



# JOURNAL

A Quarterly Publication of the Cyber Security & Information Systems Information Analysis Center

# INSIDER THREAT AND MALICIOUS INSIDER THREAT

***ANALYZE. DETER. DISCOVER. PREVENT. RESPOND.***

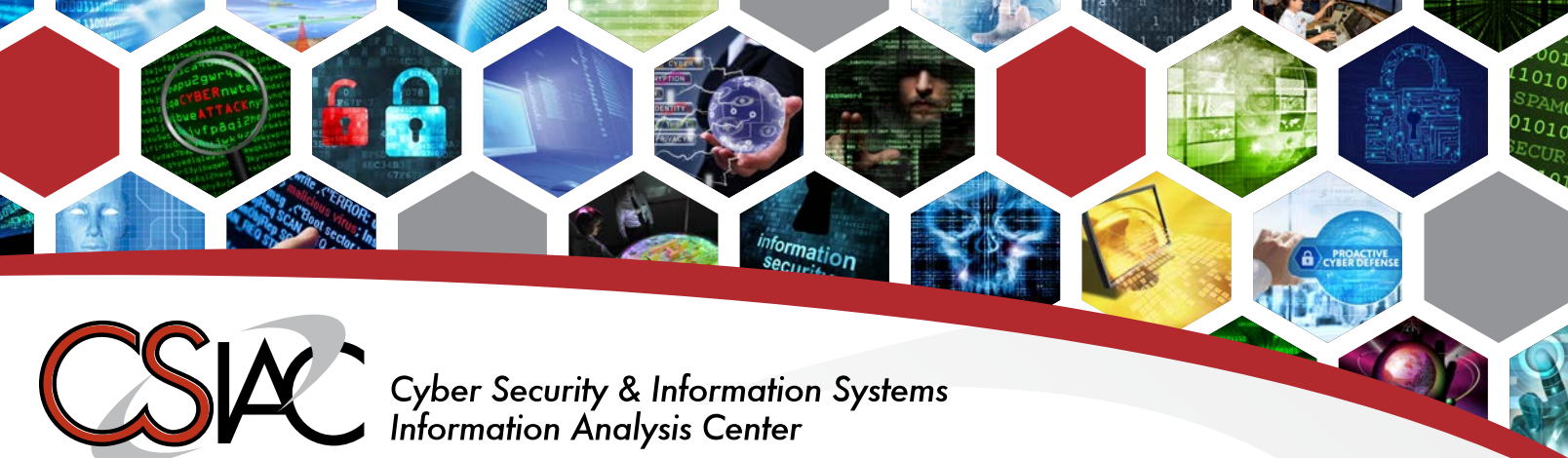


**Featuring articles from the  
SANS Technology Institute**

*most trusted* source for  
information security  
training and security  
certification in the world.



DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.



**Cyber Security & Information Systems  
Information Analysis Center**

## ABOUT THE CSIAC

As one of three DoD Information Analysis Centers (IACs), sponsored by the Defense Technical Information Center (DTIC), CSIAC is the Center of Excellence in Cyber Security and Information Systems. CSIAC fulfills the Scientific and Technical Information (STI) needs of the Research and Development (R&D) and acquisition communities. This is accomplished by providing access to the vast knowledge repositories of existing STI as well as conducting novel core analysis tasks (CATs) to address current, customer focused technological shortfalls.

## OUR MISSION

CSIAC is chartered to leverage the best practices and expertise from government, industry, and academia in order to promote technology domain awareness and solve the most critically challenging scientific and technical (S&T) problems in the following areas:

- ▶ Cybersecurity and Information Assurance
- ▶ Software Engineering
- ▶ Modeling and Simulation
- ▶ Knowledge Management/Information Sharing



The primary activities focus on the collection, analysis, synthesis, processing, production and dissemination of Scientific and Technical Information (STI).

## OUR VISION

The goal of CSIAC is to facilitate the advancement of technological innovations and developments. This is achieved by conducting gap analyses and proactively performing research efforts to fill the voids in the knowledge bases that are vital to our nation. CSIAC provides access to a wealth of STI along with expert guidance in order to improve our strategic capabilities.

## WHAT WE OFFER

We provide expert technical advice and assistance to our user community. CSIAC is a competitively procured, single award contract. The CSIAC contract vehicle has Indefinite Delivery/Indefinite Quantity (ID/IQ) provisions that allow us to rapidly respond to our users' most important needs and requirements.

Custom solutions are delivered by executing user defined and funded CAT projects.

## CORE SERVICES

- ▶ Technical Inquiries: up to 4 hours free
- ▶ Extended Inquiries: 5 - 24 hours
- ▶ Search and Summary Inquiries
- ▶ STI Searches of DTIC and other repositories
- ▶ Workshops and Training Classes
- ▶ Subject Matter Expert (SME) Registry and Referrals
- ▶ Risk Management Framework (RMF) Assessment & Authorization (A&A) Assistance and Training
- ▶ Community of Interest (COI) and Practice Support
- ▶ Document Hosting and Blog Spaces
- ▶ Agile & Responsive Solutions to emerging trends/threats

## PRODUCTS

- ▶ State-of-the-Art Reports (SOARs)
- ▶ Technical Journals (Quarterly)
- ▶ Cybersecurity Digest (Semimonthly)
- ▶ RMF A&A Information
- ▶ Critical Reviews and Technology Assessments (CR/TAs)
- ▶ Analytical Tools and Techniques
- ▶ Webinars & Podcasts
- ▶ Handbooks and Data Books
- ▶ DoD Cybersecurity Policy Chart

## CORE ANALYSIS TASKS (CATS)

- ▶ Customer tailored R&D efforts performed to solve specific user defined problems
- ▶ Funded Studies - \$1M ceiling
- ▶ Duration - 12 month maximum
- ▶ Lead time - on contract within as few as 6-8 weeks

## CONTACT INFORMATION

266 Genesee Street  
Utica, NY 13502

1 (800) 214-7921

info@csiac.org

/DoD\_CSIAC

/CSIAC

/CSIAC



# ABOUT THE JOURNAL OF CYBER SECURITY AND INFORMATION SYSTEMS



## JOURNAL EDITORIAL BOARD

**SANS Technology Institute**  
CSIAC Editorial Board Member

**RODERICK A. NETTLES**  
Managing Editor  
Quanterion Solutions Inc., CSIAC

**MICHAEL WEIR**  
CSIAC Director  
Quanterion Solutions Inc., CSIAC

**ERIC PATTERSON**  
Executive Director  
SANS Technology Institute

**DR. JOHANNES ULRICH**  
Director of Internet Storm Center  
SANS Technology Institute

**DR. DAVID T. VACCHI**  
Director of Curriculum Development  
SANS Technology Institute

**LEAH TREMAGLIO**  
Student Advisor/Writing Instructor  
SANS Technology Institute

**DR. PAUL B. LOSIEWICZ**  
Senior Scientific Advisor  
Quanterion Solutions Inc., CSIAC

**SHELLEY STOTTLAR**  
Graphic Designer  
Quanterion Solutions Inc., CSIAC

## ABOUT THIS PUBLICATION

The **Journal of Cyber Security and Information Systems** is published quarterly by the Cyber Security and Information Systems Information Analysis Center (CSIAC). The CSIAC is a Department of Defense (DoD) Information Analysis Center (IAC) sponsored by the Defense Technical Information Center (DTIC) and operated by Quanterion Solutions Incorporated in Utica, NY.

Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the CSIAC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the CSIAC, and shall not be used for advertising or product endorsement purposes.

## ARTICLE REPRODUCTION

Images and information presented in these articles may be reproduced as long as the following message is noted:

***“This article was originally published in the CSIAC Journal of Cyber Security and Information Systems Vol.6, No 1”***

In addition to this print message, we ask that you notify CSIAC regarding any document that references any article appearing in the CSIAC Journal.

Requests for copies of the referenced journal may be submitted to the following address:

### Cyber Security and Information Systems

266 Genesee Street  
Utica, NY 13502

Phone: 800-214-7921

Fax: 315-732-3261

E-mail: [info@csiac.org](mailto:info@csiac.org)

An archive of past newsletters is available at <https://www.csiac.org/journal/>.

*To unsubscribe from CSIAC Journal Mailings please email us at [info@csiac.org](mailto:info@csiac.org) and request that your address be removed from our distribution mailing database.*

## Journal of Cyber Security and Information Systems

### Insider Threat and the Malicious Insider Threat

*Analyze. Deter. Discover. Prevent. Respond.*

Introduction: Insider Threat and the Malicious Insider Threat – Analyze. Deter. Discover. Prevent. Respond .....	4
Extensions to Carnegie-Mellon University’s Malicious Insider Ontology to Model Human Error .....	6
Detect, Contain and Control Cyberthreats .....	13
Compliant but not Secure: Why PCI-Certified Companies Are Being Breached .....	18
Accessing the Inaccessible: Incident Investigation in a World of Embedded Devices .....	26
Offensive Intrusion Analysis: Uncovering Insiders with Threat Hunting and Active Defense .....	40

# INTRODUCTION: INSIDER THREAT AND THE MALICIOUS INSIDER THREAT – Analyze. Deter. Discover. Prevent. Respond.

By: Michael Weir, CSIAC Director, and  
Roderick A. Nettles, CSIAC Deputy Director/Managing Editor

**B**uilding a quarterly journal that spans broad topical and technical themes can be challenging, and the selection of articles for any one journal intimidating. Over the last five years CSIAC has published special issues on military research laboratories (Volume 5 Number 1; Volume 4 Number 1), focused in on particular relevant technical thrusts (i.e., Serious Games M&S, Volume 5 Number 4, December 2017), and operational considerations (i.e., SCADA, Volume 1 Number 3). This quarter, the CSIAC Journal presents five articles which represent different perspectives on Insider Threat and approaches to understand and remediate that threat. Due to the cost of reproduction and distribution, we are releasing the print journal with the first four articles, and incorporating into the journal a reference to the longer and more complex fifth article available online at [CSIAC.ORG](http://CSIAC.ORG). All five articles are included in the PDF version of the journal available online.

***In this journal we are proud to identify and include work by two organizations with a long history of research and good counsel regarding Insider Threat – the Software Engineering Institute (SEI) at Carnegie Mellon University and the SANS Technology Institute.***

Any collaboration between people in a group requires a certain degree of trust to be successful. Whether in financial, political, military, or social situations, the ability to trust those around you is a primary enabling factor to success. Misuse of that trust to gain advantage for purposes counter to the group's success can also be a primary factor in the group's failure. For the last few decades in the cybersecurity realm, the term "Insider Threat" has been used to identify individuals or entities that misuse some level of trust gained within an organization to adversely use information or information systems to the detriment of the organization. The designation is somewhat broad, encompassing intentional and unintentional actions, individuals and groups of people, even human and machine/computer activities. Approaches to the remediation of the Insider Threat are also quite broad, with current best practice combining several to achieve the best results. Physical, technical, behavioral, policy, and process means are all parts of an effective Insider Threat program.

When any concept or technology becomes widely relevant, it begins to differentiate into sub-components on its path to full maturity. New and innovative approaches leverage and augment the foundational ideas that generated the original concepts, frequently evolving to new areas and spawning their own subcomponents. Insider Threat remediation research has made that journey, growing in relevance and maturity, and many alternative paths evolved from those foundational ideas as the methods and technology behind information management (and the methods and technology available to Insider Threat actors) have become more powerful and complex. One piece of the puzzle has remained a constant – the human aspect.



The American origins of Insider Threat conduct go back at least to 1775. Benjamin Church was a British Loyalist and trusted insider who had access to important Colonial letters by virtue of his position. He diverted key messages to British general Sir Thomas Gage in an attempt to undermine American military movements<sup>1</sup>. The same human motivations that drove his actions have been repeated over and over again in the last two centuries, using different methods and technologies to access and misuse critical information. In the late 1980's, the CIA initiated Project Slammer in an attempt to gather the most current and relevant information from captured insider spies to discern the primary influencers that enabled their conduct. At the end of that heavily redacted 1990 report<sup>2</sup>, quote:

*“Subjects almost invariably conceive of committing espionage after they are in a position of trust. While initial screening continues to be important, focusing on update and monitoring procedures seems increasingly worthwhile.”*

In a Counterintelligence Trends document from 1993<sup>3</sup> summarizing the overall Project, it states clearly that none of the people studied intended to spy at the point they were granted access to information.

With that firmly in mind, this special issue will focus on the “Insider Threat and the Malicious Insider Threat” that pose unique security challenges to all organizations due to their knowledge, proficiencies, and authorized access to information systems.

**How do you interpret people's behavior in the context of the Insider Threat?** The next article identifies and amplifies concepts associated with a core concern of many involved with Insider Threat – *what about the unintentional insider?* Professor Coffey expands on the Software Engineering Institutes' (SEI's) Insider Threat Ontology to recommend some ways to incorporate non-malicious behavior within that construct, and provides an exemplar of how it might be used.

1- Benjamin Church, probably the first Surgeon General of the US, provided information to the British prior to the Battle of Lexington, reference here: <http://clements.umich.edu/exhibits/online/spies/people.html#church>

2 - Project Slammer Interim Report, 12 April 1990, redacted and declassified version available here: [https://www.cia.gov/library/readingroom/docs/DOC\\_0000218679.pdf](https://www.cia.gov/library/readingroom/docs/DOC_0000218679.pdf)

3 - “Counterintelligence Trends”, DCI Counterintelligence Center, January 1993, page 10; approved for release March 2002, available here: <https://cryptome.org/2013/06/cia-why-spy.pdf>

**If you can't stop the Insider, how do you mitigate the effects?** The following article identifies a truth about compromise (with enough effort, virtually any organization can be compromised) and then proposes methods for most effectively mitigating the effects of compromise. Dr. Cole proposes best-practice methodologies for Detect, Contain and Control with an emphasis on the Insider Threat.

**How do you integrate policy and compliance with an effective Insider Threat program?** A very different perspective is provided by Christian Moldes in his article on the policy-level components of an effective Payment Card Industry (PCI) compliance program, identifying the effective integration of the objectives of compliance with the organization's organic actions/processes in place to assure protection of information assets.

**What about the threat of “Insider Hardware” that isn't even a person?** With the Internet of Things (IoT) becoming a component part of any organization, what about the threat of embedded hardware inside your organization? Eric Jodoin provides a very detailed example of revealing an embedded devices' information flow using serial port access. It is illuminating both for the ability to access embedded information streams and the reasoning process that can provide insight into how embedded devices can be used in an insider scenario.

**How do we get better at finding Insider Threats?** Matthew Hosburgh suggests a more contemporary method for actively identifying Insider Threat actors – applying the concepts of Threat Hunting to the problem. Involving people more actively in the hunting of Insider Threat actors using current Threat Hunting tools and techniques ratchets up the capability to find and remediate potential problems. This article also capitalizes on the Insider Threat Ontology from the SEI and identifies insertion points for the Threat Hunting methods.

We hope that this combination of articles across a broad spectrum of Insider Threat remediation techniques and analyses will help you go beyond the basic, first-order effects of traditional Insider Threat tools and ideas and begin to reason about the wider aspects of how people, technology and policy can combine more coherently to analyze, deter, discover, and ultimately prevent such activities from occurring.



# EXTENSIONS TO CARNEGIE-MELLON UNIVERSITY'S MALICIOUS INSIDER ONTOLOGY TO MODEL HUMAN ERROR

By: Dr. John W. Coffey, Department of Computer Science, University of West Florida

**Researchers at Carnegie-Mellon University have created an “Insider Threat Ontology” as a framework for knowledge representation and sharing of malicious insider cases.** *The ontology features rich constructs regarding people who take malicious actions to compromise or exploit cyber assets. However, modeling end-user error was outside the scope of the CMU work. The current work enumerates extensions to the CMU ontology to model end-user and system administrator error. Specifically, additions to the inheritance lattice of actors is presented and additional types of actions pertaining to human error are described. The article concludes with an example of the use of these extensions to model a case drawn from the Privacy Rights Clearinghouse database of data breaches.*



## 1. Introduction

A truly alarming percentage of cybersecurity problems occur because of human mediation leading to the compromise or circumvention of technological safeguards. A variety of people within an organization can play a role in an attack. Malicious insiders who are either system administrators or end users are responsible for a truly significant amount of havoc and loss. Additionally, however, improperly trained or insufficiently vigilant system administrators commit a range of errors. End users with no malicious intent are also a significant source of error that leads to damaging and ultimately unnecessary loss.

The malicious insider problem has long been recognized as a daunting challenge in cybersecurity. Disgruntled employees who leave organizations with sensitive data, and current employees who perceive the chance to profit by selling secrets are just two of the

many forms of malicious insider behavior. Accordingly, researchers at Carnegie Mellon University's Software Engineering Institute have created a highly articulated ontology [1] to standardize the vocabulary and to map relationships among entities for the formal description of malicious insider attacks. The ontology provides a well-thought out knowledge representation scheme that can be used to share information in a standardized form and to build reasoning systems pertaining to the domain.

The CMU ontology is well-designed and comprehensive with regard to knowledge representation of malicious insider actions, events, assets, and information. However, it was not intended to include the modeling and representation of knowledge pertaining to errors made by people. The purpose of this paper is to propose extensions to the CMU Malicious Insider Ontology to enable explicit modeling of human error in cybersecurity breaches.

The remainder of this paper contains a description of literature pertaining to human error in cybersecurity breaches, and a set of extensions to CMU's ontology that would permit modeling of human error-mediated events. After a discussion of the CMU model, the proposed extensions are presented and discussed. Use of the extensions is illustrated by formally representing a narrative pertaining to a real-world data breach documented in the Privacy Rights ClearingHouse [17] database, a database that documents 5,200 data breaches made public since 2005. The paper concludes with some lessons drawn from the exercise.

## 2. Human Error in Cybersecurity Vulnerabilities

Kharif [18] reports that 2016 was the worst year in history for data breaches. The Identify Theft Resource Center [19] recorded in excess of 1,000 breaches in 2016, exhibiting an increase of 40% from 2015. According to a report by IBM, 95% of all security breaches are mediated in some degree by human beings [2]. Daugherty [21] summarizes BakerHostetler's 2015 and 2016 Data Security Incident Response Reports. The 2015 report reads that 37% of incidents involved human actions or errors. This characterization was of systems administrator error and end-user error. Daugherty states that successful phishing/malware attacks contributed to 25% of data breaches. His summary suggests human error playing a role in 62% of all incidents. BakerHostetler's statistics were based upon a smaller sample than IBM's data, but the message from both is the same – human error is an extremely important proximate cause of security breaches.

Howarth [3] describes a range of human errors often involving people inside organizations who do dangerous things either accidentally or deliberately. Major categories of human error include the inadvertent exposure of sensitive data, creating conditions that allow the introduction of malware into mission-critical systems, and creating conditions that allow theft of intellectual property or sensitive information.

Howarth concludes that organizations that implement strong technological security procedures still often pay insufficient attention to human sources of vulnerability, including errors made by system administrators. He strongly advocates for enhanced security training to decrease human error. Armerding [8] cites a report that indicates that 56% of workers who use the Internet on their jobs receive no security training at all. While malicious insiders remain a significant threat to cybersecurity, it is clear that enormous problems arise from people with no malicious intent performing dangerous behaviors or being tricked into compromising sensitive information.

Social Engineering attacks are those that involve tricking people into making errors. One of the most common forms of social engineering attacks is phishing, and surprisingly, phishing attacks still frequently succeed. Modern phishing attacks are much more sophisticated than in earlier times and typically aim to install malware. Spear phishing attacks are highly sophisticated, being based upon emails that appear to be from trusted sources or businesses with which the target of the attack has interactions. Phishing attacks are so common and so successful that a worldwide working group, [antiphishing.org](http://antiphishing.org) [20], has formed to foster research, data exchange and public awareness of the problem.

A surprisingly common problem arises from people simply moving sensitive data around via unsafe technologies. People send documents to the wrong recipient through email, carry sensitive information on jump drives and place sensitive documents on insecure file sharing sites. Verizon reports that 63% of confirmed data breaches were facilitated by the use of legitimate passwords that were weak, default or stolen. Point of sale attacks frequently occur and they are often caused by people who use point of sale machines for other uses including web surfing and email [7]. Verizon estimates that only 3% of phishing attacks are reported by the targets of the attacks. A quick scan of the author's spam file reveals 5 emails with suspicious attachments out of the 215 emails currently in the folder.

***Malicious  
insiders who are  
either system  
administrators  
or end users are  
responsible for a  
truly significant  
amount of havoc  
and loss.***

System Administrators are responsible for significant problems as well. Barrett, Chen, and Maglio [23] discuss the increasing challenges of monitoring and maintaining increasingly complex systems, and the costs associated with deficient processes. Fulp et al [24] describe errors in systems configuration such as firewall maintenance, patch management, and failures of signature-based intrusion detection, as critical problems. Other system administrator-mediated problems include lack of access control management as end users change roles within an organization or when they leave organizations. Seemingly simple-to-rectify problems such as ineffective patch-management become more significant as system administrators are called upon to manage more and larger systems.

## 3. The CMU Malicious Insider Ontology

The CMU Malicious Insider Ontology was created to provide a standard means to formally represent cases of malicious activity inside organizations [1]. Their concern is to model people who were formerly or are currently associated with an organization who had privileged access that they deliberately used in a fashion that negatively impacted the organization. Insider status pertained to anyone with privileged access including contractors and people working for trusted business partners of the victim organization.



The authors of the CMU ontology quote Gruber [11] who described an ontology as a “coherent set of representational terms, together with textual and formal definitions that embody a set of representational design choices.” In other words, an ontology provides a standardized vocabulary for objects, actions and relationships among its constituent parts that affords knowledge representation in a sharable and machine-processable form.

The authors of the CMU Malicious Insider Ontology chose to utilize Web Ontology Language 2 (OWL 2) [12] constructs to implement the ontology because it is a standard of the World Wide Web Consortium and has many support tools. They performed a round of concept mapping [13] to assess the scope of the project. They identified the following five base classes:

- › Actor
- › Action
- › Event
- › Asset
- › Information

They chose to model `Action` and `Event` as separate classes, making the distinction that actions are observable occurrences and events are inferred to have occurred because of actions. Additionally, an event might be comprised of multiple actions. For instance, their taxonomy includes high-level concrete actions such as `DigitalAction`, `FinancialTransactionAction`, and `JobChangeAction`, and modifier actions such as `SuspiciousAction` (an activity that might raise concerns about malicious behavior). Modifier actions are used to qualify the concrete actions. For instance, an `EmailAction` (a type of `DigitalAction`) might also be a `SuspiciousAction`. The eleven `Event` classes include `DataExfiltrationEvent` (unauthorized removal of data from a computer), and `JobOfferEvent` (getting an offer to work somewhere else). The ontology has extensive representations pertaining to people leaving or changing jobs, since such events are often the starting point for a variety of malicious activities.

The `Asset` and `Information` classes comprehensively enumerate a vocabulary for their respective domains. Assets are characterized

as the targets of malicious actions. The CMU ontology includes a temporal component for the representation of a series of actions and events that occur over time, and they incorporate Peterson’s [14] SpaceTime Ontology to model events occurring over time.

## 4. Extensions for End-user Error

The CMU ontology features a total of 124 classes. It has wide-ranging modeling constructs for `Asset` and `Information` that are applicable to any cybersecurity incident. The `Actor` class contains two broad subtypes: `Organization` and `Person`. The `Action`, and `Event` classes appear to be focused on malicious behaviors by insiders and are not meant to provide significant capabilities to model human error. In the next sections, extensions to these three classes are presented to afford more fine-grained modeling of actors and modeling of a variety of human errors committed both by end users and system administrators.

### 4.1 Extensions to Actor

The design decision to have the `Actor` class have only two subclasses is understandable because the intent of the ontology is to model malicious behaviors by insiders. Consequently, instances of the `Person` class would typically be the bad actor(s) in the incident. Since it is possible that innocent people might be duped into aiding the bad actor, greater articulation of these classes affords a more fine-grained modeling capability. Figure 1 contains an extended class lattice for the `Actor` class. The yellow ellipses are classes from the CMU ontology, and the gray ellipses are the extensions proposed here.

As can be seen in Figure 1, a differentiation is made between a malicious actor and one who does not have malicious intent. Additionally, the notion is presented that either people in charge of systems or end users may be malicious or not. These extensions make it possible to provide fine-grained descriptions of human error either on the part of those who are responsible for administering systems or for those who use systems. Note that the `is_one_of` relationship disambiguates the superclass-subclass relationships, creating a logical or rather than a logical and.

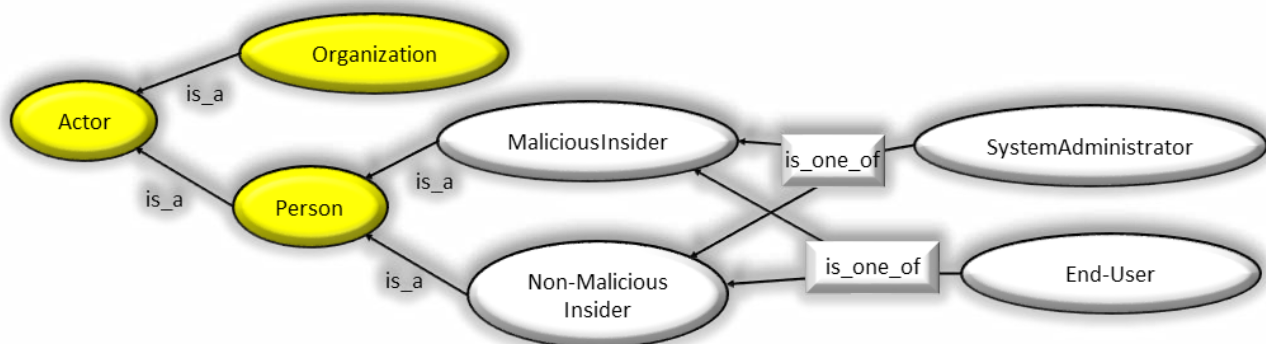


Figure 1: The Inheritance Hierarchy for Actor from the CMU Ontology with Extensions for Human Error. - (Source: Author)

#### 4.2 Extensions to Action

Creating explicit constructs to represent erroneous actions might be achieved in any of several ways. One possibility is to have class *ErrorAction* as an extension of the *ActionModifier*. Under this approach, accidentally sending an email with sensitive information to the wrong recipient would involve a *DigitalAction.EmailAction* modified by *ErrorAction*. Another alternative is to have *ErrorAction* as a separate top-level action. Since the other actions are deliberately performed by a malicious actor and *ErrorAction* instances are not, it is reasonable to model *ErrorAction* as its own top-level *Action*. A third alternative is to consider all error actions to be special cases of *UnauthorizedAction*. The reason for creating the separate higher-level class instead of subclassing *UnauthorizedAction* is that in the malicious insider taxonomy, an unauthorized action would be deliberate and in the error taxonomy it is explicitly represented to be inadvertent.

An initial set of *ErrorAction* types has been identified as indicated in Figure 2. These categories were culled from the literature on end user error and do not constitute an exhaustive list. It should be noted that ontological modeling routinely requires multiple inheritance. Some generalization-specialization relationships are excluded from Figure 2. For example, a *BadEmailSendAction* might be an *ErrorAction* and a *FinancialTransactionAction*.

#### 4.3 Extensions to Event

As previously stated, the CMU ontology models eleven events with a clear focus on deliberate, malicious acts. Events include *SystemModificationEvent*, *DataDeletionEvent*, *FraudEvent*, *DataExfiltrationEvent*, *SabotageEvent*,

*TheftEvent*, etc. The intent in creating these Event types is clear – they subsume potentially several actions. For instance, a *DataExfiltration* event might involve copying data into an email and emailing the data to an entity that competes with the victim organization. In order to parallel the morphology of the CMU work, a subsuming notion of an *ErrorEvent* is created. An *ErrorEvent* is disjoint from six of the eleven Event categories in the CMU ontology (for instance a *MasqueradingEvent* which is necessarily deliberate in nature) but overlapping several such as *DataDeletionEvent* which might be an accidental or malicious event. The fact that such extensions can be seamlessly integrated into the CMU ontology suggests that the ontology is well-structured.

### 5. An Example Application of the Extended Ontology

The CMU report includes a methodology to model textual descriptions of incidents, and several examples of doing so. Commonly used symbology includes representing classes as labeled yellow ellipses and the use of purple diamonds to represent instances of classes. Arrows are used to represent relationships between classes and instances. The directionality of the arrows removes any ambiguity regarding the nature of the relationships. The scheme that is utilized in the CMU examples is the same as the one produced by the Protégé tool developed at Stanford [22], a tool that supports OWL2 Web Ontology language. The method that is used to create the formalization is algorithmic:

- › Identify the main Actors
- › Model Actors' actions
- › Establishing basic relationships.
- › Model IT Infrastructure
- › Connect Infrastructure to Actors
- › Model IT actions

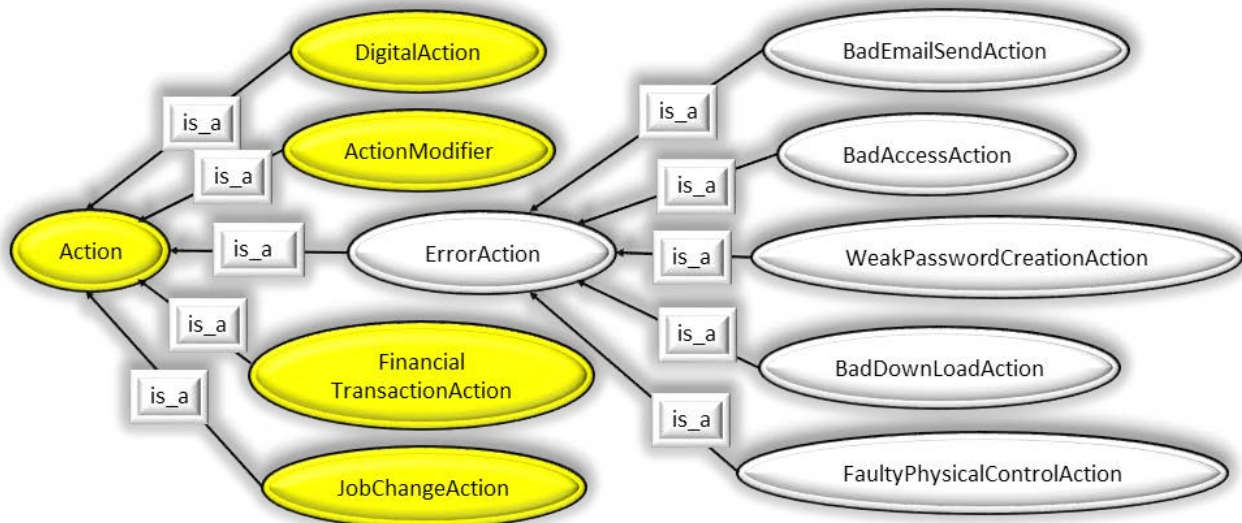


Figure 2: The Inheritance Hierarchy for Action from the CMU Ontology, and Extensions for Human Error. - (Source: Author)

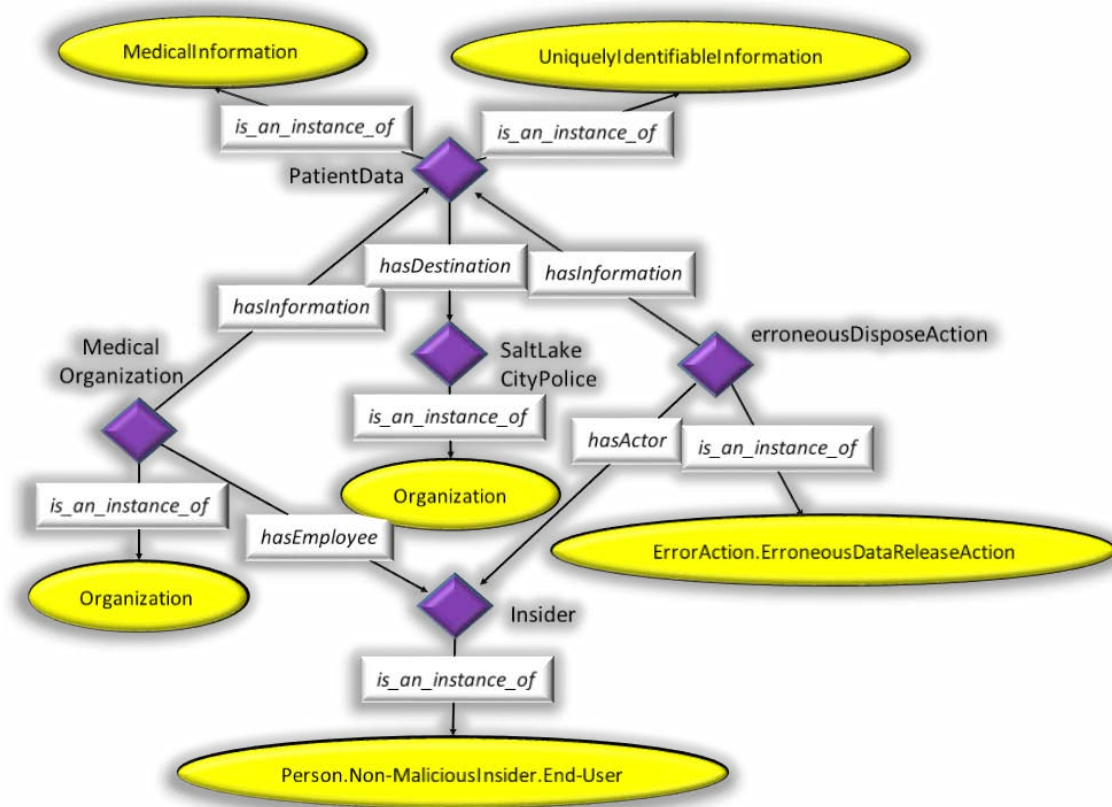


Figure 3: Representation of an End-User Error Using the Extended Ontology. - (Source: Author)

The process described in the CMU work was used in the current work. The process is well thought out and logical and its use led to a straight-forward representation of the example case presented next.

Figure 3 illustrates how a data breach incident is formally modeled utilizing elements of the Insider Threat Ontology created at CMU and proposed extensions to that ontology. As stated, the methodology described in the CMU report was used to create the representation of the case. The representation evolved through multiple steps. As the process is thoroughly described in the CMU document, only the final result is presented rather than the step-by-step evolution of the representation. The following passage, which is represented in Figure 3 using the ontology, was taken verbatim from the Privacy ClearingHouse database [17] that documents 5,000 data breaches.

*Names, credit card numbers, Social Security numbers were found in a dumpster. A man was throwing away some stuff in a dumpster and found it was chock full of medical records. "There's everything in there from canceled checks to routing numbers," he said. Salt Lake Police packed away perhaps twenty boxes of papers, and said they would protect the documents, as they dug into the matter.*

## 6. Discussion

The base classes of the CMU Insider Threat Ontology provide extensive modeling capability for human actions leading to cybersecurity breaches. The extensions to the malicious insider ontology proposed here proved to be sufficient for the formalization of the case chosen from the Privacy ClearingHouse database. As would commonly occur using the Insider Threat Ontology, the current case illustrates the use of multiple inheritance. In Figure 3, it can be seen that *patientData* is an instance both of *MedicalInformation* and of *UniquelyIdentifiableInformation*, both of which are part of the CMU ontology. This representation was necessary since both patient medical records and social security numbers were compromised in the case being modeled. As discussed before, multiple inheritance schemes are common in ontology construction. The Asset and Information class hierarchies in the CMU ontology are well formed, comprehensive. They were used without modification for the modeling of this particular end user error case.

The model in Figure 3 utilizes a common convention (in the form of a dot notation) for superclass-subclass representation. Some form of representation akin to dot notation is commonly used to provide fully-qualified names in computer programming languages. For instance, the *Person.Non-MaliciousInsider.End-User* characterization disambiguates the end-user from one who has malicious intent.

## 7. Conclusions

The authors of the CMU report presenting the Insider Threat Ontology correctly state that the lack of a broadly accepted, standard, formal representation for knowledge pertaining to the field of cybersecurity is a sign of immaturity in the field. Their ontology represents an important step forward in the resolution of this deficiency. The extensions proposed in this article provide a means to model accounts of cases involving human error in addition to deliberate malicious actions. These extensions also permit more detailed characterization of the nature of the actor as a technical person or end user. It is inherently difficult to model the real world formally, and consequently, object modeling is always an iterative process. Most assuredly, additional refinements to the extensions suggested here are called for. Nevertheless, these extensions are a first step toward enabling fine-grained representation of security breaches involving human error.

## REFERENCES

- [1] Costa, D.L., Albrethsen, M.J., Collins, M.L., Perl, S.J., Si-lowash, G.J., and Spooner, D. An Insider Threat Indicator Ontology. Technical Report CMU/SEI-2016-TR-007. Online. Available: [https://resources.sei.cmu.edu/asset\\_files/TechnicalReport/2016\\_005\\_001\\_454627.pdf](https://resources.sei.cmu.edu/asset_files/TechnicalReport/2016_005_001_454627.pdf).
- [2] IBM Security Services 2014 Cyber Security Intelligence Index. Online. Available: [https://media.scmagazine.com/documents/82/ibm\\_cyber\\_security\\_intelligenc\\_20450.pdf](https://media.scmagazine.com/documents/82/ibm_cyber_security_intelligenc_20450.pdf)
- [3] F. Howarth. The Role of Human Error in Successful Security Attacks. Online. Available: <https://securityintelligence.com/the-role-of-human-error-in-successful-security-attacks/>
- [4] A. Glaser. Here's What We Know about Russia and the DNC Hack. <https://www.wired.com/2016/07/heres-know-russia-dnc-hack/>
- [5] Verizon. 2013 Data Breach Investigations Report. Online, available: [http://www.verizonenterprise.com/resources/reports/rp\\_data-breach-investigations-report-2013\\_en\\_xg.pdf](http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf)
- [6] Hummer, L. Security Starts with People: Three Steps to Build a Strong Insider Threat Protection Program Online, Available: <https://securityintelligence.com/security-starts-with-people-three-steps-to-build-a-strong-insider-threat-protection-program/>
- [7] Coffey, J. W., Baskin, A. and Snider, D. Knowledge Elicitation and Conceptual Modeling to Foster Security and Trust in SOA System Evolution. In El-Sheikh, E., Zimmerman, A., and Jain, L. (Eds), *Emerging Trends in the Evolution of Service Oriented and Enterprise Architectures*. 2016. Springer.
- [8] J. Grunzweig. Understanding and Preventing Point of Sale Attacks. Online. Available: <http://researchcenter.paloaltonetworks.com/2015/10/understanding-and-preventing-point-of-sale-attacks/>
- [9] A. Hern. What is Dridex, and how can I stay safe? Online, Available: <https://www.theguardian.com/technology/2015/oct/14/what-is-dridex-how-can-i-stay-safe>
- [10] T. Armerdeing. Security training is lacking: Here are tips on how to do it better. Online, Available: <http://www.csoonline.com/article/2362793/security-leadership/security-training-is-lacking-here-are-tips-on-how-to-do-it-better.html>
- [11] Gruber, T.R. The role of common ontology in achieving sharable, reusable knowledge bases. In Allen, J. A., Fikes, R., and Sandewall, E. (Eds.) *Principles of Knowledge Representation and Reasoning*. Proceedings of the Second International Conference. San Mateo, CA: Morgan Kaufmann, 1991.
- [12] W3C. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). Online, Available: <https://www.w3.org/TR/owl2-syntax/>
- [13] Novak, J.D. and Gowin. *Learning How to Learn*.
- [14] Peterson, E. SpaceTime Ontology. Online. Available: <http://semantic.org/Ontology/OntDef/Cur/SpaceTime.owl>
- [15] Mundie, D. How Ontologies can Help Build a Science of CyberSecurity. Online, Available: <https://insights.sei.cmu.edu/insider-threat/2013/03/how-ontologies-can-help-build-a-science-of-cybersecurity.html>
- [16] Mundie, D., and McIntire, D. The Mal: A Malware Analysis Lexicon. Online, Available: <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=40240>
- [17] Privacy Rights Clearinghouse. Data Breaches. Online. Available: <https://www.privacyrights.org/data-breaches>
- [18] Kharif, O. 2016 Was a Record Year for Data Breaches. Online. Available: <https://www.bloomberg.com/news/articles/2017-01-19/data-breaches-hit-record-in-2016-as-dnc-wendy-s-co-hacked>
- [19] ITRC. The Identity Theft Resource Center. Online. Available: <http://www.idtheftcenter.org/>
- [20] APWG. APWG: Unifying the Global Response to Cybercrime. Online, Available: <http://www.antiphishing.org/>
- [21] W. R. Daughterty. Human Error Is to Blame for Most Breaches. Online, Available: <http://www.cybersecuritytrend.com/topics/cyber-security/articles/421821-human-error-to-blame-most-breaches.htm>
- [22] Protégé. What is an Ontology? Building and Inference Using The Stanford Protege tool Part I. Online. Available: <https://www.youtube.com/watch?v=1IQScWqzPw>
- [23] System Administrators are Users Too: Designing Workspaces for Managing Internet-Scale Systems. Proceedings of CHI2003. pp 1068- 1069. April 5-10, 2003, Ft. Lauderdale, Florida, USA. ACM 1-58113-637-4/03/0004.
- [24] Fulp, E. W., Gage, D., John, D., McNiece, M., Turkett, W., and Zhou, X. An Evolutionary Strategy for Resilient Cyber Defense. Proceedings of the 2015 IEEE Global Communications Conference. DOI: 10.1109/GLOCOM.2015.7417814

## ABOUT THE AUTHOR

**Dr. John W. Coffey** holds a B.S. in Psychology from the College of William and Mary (1971), a B.S. in Systems Science (1989), an M.S. in Computer Science/Software Engineering (1992), and an Ed.D. with an emphasis in Computer Science (2000) from the University of West Florida (UWF). He was one of the first members of the Institute for Human and Machine Cognition (IHMC) and he worked with that organization for many years. He has been in the Department of Computer Science at the University of West Florida since 1992, starting as a Lecturer and working his way up to his current rank of Professor. He has published more than 100 refereed journal articles, book chapters, technical reports and conference proceedings. His research interests include knowledge elicitation, representation, and modeling, human-mediated cybersecurity breaches, and computer science education.



# DETECT, CONTAIN AND CONTROL CYBERTHREATS

By: Dr. Eric Cole, PhD, SANS Institute Certified Instructor and Fellow

**Today, every organization is a target** and attackers can compromise any organization. Large-scale compromises used to be a surprise, but now they are a reality that is often accepted. The means, methods and techniques that adversaries use to target and ultimately compromise organizations have caused a shift in mind-set. It is not a matter of if an attacker will compromise an organization, but when an attack will occur.

Although prevention is ideal, not all attacks can be prevented, making compromise inevitable. Therefore, a better approach to security is timely detection of the attack—detection that will contain and control the damage. Organizations that cannot detect and control the damage of an attack will cease to exist, while those that can implement effective security to minimize the impact of attacks will be the successful entities of the future.

In recognizing that attackers will succeed, the goal becomes minimizing the exposure and damage. This correlates into two key metrics:

**Dwell Time.** This includes the time from when someone clicks (you are compromised) until the time the malware is no longer effective, whether that be by blocking command and control so it cannot communicate or by taking the compromised box(es) off the network. This directly relates to damage, because the longer a system is compromised, the bigger the impact. This is a very similar approach to disease, where the goal is prevention and early detection, because the longer the disease can exist within a body, the more damaging and lethal it is to the individual. Controlling dwell time means early detection with appropriate response.

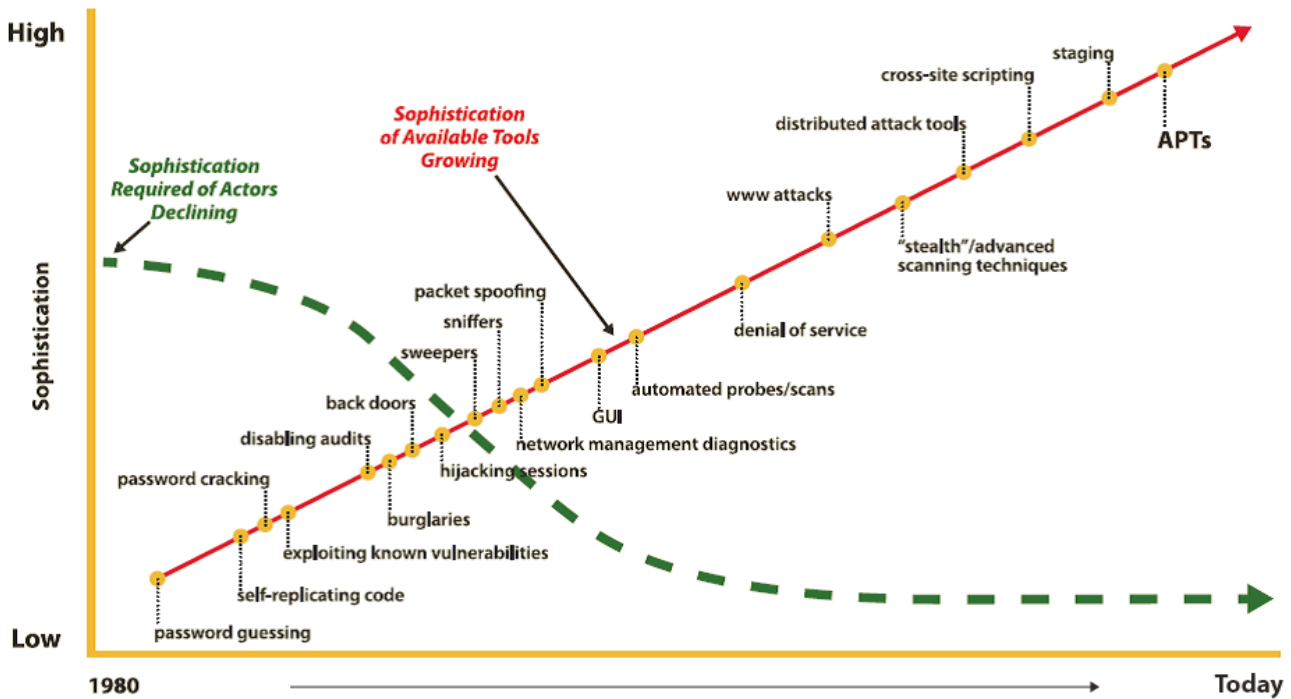


Figure 1: Evolution of Cyberthreats - (Source: Author)

**Lateral movement.** Closely tied to dwell time and in a fashion similar to cancer spreading in a body, an adversary will try to cause more damage by trying to move within an organization, compromising as many systems as possible.

As organizations build mature security programs, it is critical that they detect attacks early (reduce dwell time) and control the damage (limit lateral movement).

When designing, deploying and building networks, organizations must assume that the networks will be compromised. Trying to fix every vulnerability within an organization is an unreasonable goal, but prioritizing mitigation efforts based on known risks and high value targets can lead to success.

Organizations also need to focus on two key characteristics of risk: likelihood and impact, because not all threats are equal. An organization should prioritize threats that are likely to occur and in the process, cause great damage.

To help keep your organizations aligned on containment and control, before you spend a dollar of budget or an hour of time on security problems, you should always ask three questions:

- What are my high-value targets – data, machines, and personnel?
- What the risks if these high value targets are comprised?
- What are the most cost-effective ways of reducing risks?

The answers to these questions will help you prioritize risks and deploy appropriate defenses. This paper will help define strategies and tactics for this approach.

***As organizations build mature security programs, it is critical that they detect attacks early (reduce dwell time) and control the damage (limit lateral movement).***

Over the past several years, the means and methods that attackers use to compromise an organization have changed dramatically. In the past, attacks were visible and opportunistic, targeting low-hanging fruit and operating on a large scale. Therefore, many of the security technologies and solutions developed in response to such attacks focused on looking for specific ways an attacker worked, typically through signature based detection. However, today’s organizations are grappling with advanced threats that are stealthy, targeted and data-focused, rendering traditional security defenses ineffective.

Traditional attacks targeted servers in an organization’s so-called demilitarized zone (DMZ), or perimeter network—typically hosting outward-facing services such as email and web—and exploited vulnerabilities in those systems. Even if attackers were able to compromise such a server, the machine was isolated on a separate network and did not contain sensitive data. Today’s attackers target insiders within a network and employ victims as points of compromise.

Although this sounds like the work of sophisticated attackers, in reality the tools have become more capable, while the people behind the tools no longer need to be experts to take advantage of vulnerable systems. The increasing sophistication of cyberthreats is depicted in Figure 1.

When people think about computer attacks, they often visualize them as external threats. Although this is often true, it is important to differentiate between the source of a threat and the cause of damage. Although the source of most threats may be external, internal threats are increasingly real and on the minds of security analysts and IT managers. The 2015 SysAdmin, Audit, Network, and Security (SANS) survey on insider threats showed that threats from malicious and negligent employees concern most organizations: 74 percent of respondents cited employees, rather than contractors, as their greatest headache.<sup>1</sup>

When people hear insider threat, many initially think of malicious threats such as an embezzler or data thief—someone within the organization who deliberately and maliciously wants to cause harm. Although that certainly is one form of insider threat, more likely threats come from accidental insiders, people an attacker tricks or manipulates into doing something they normally would not do if they knew the true intent. Modern security solutions must address such accidental insiders.

## Current Challenges

Organizations that focus on external prevention continue to struggle with security. Although they can prevent some attacks, many others can easily slip past preventive measures and compromise internal systems. If an organization cannot detect an attack in a timely manner and limit the dwell time, the damage an attack causes will be significant. Modern IT security means putting more focus on internal detection and controlling the damage. As attacks continue, organizations are willing to invest more money toward the security budget, but finding the correct types of skilled personnel remains one of the most significant challenges. Given this constraint, the goal for almost every IT department is to automate security and present information in an intuitive, easy-to-use manner that facilitates timely and appropriate action to mitigate risks within the organization. Automating this processing and analysis with proper tools allows the security operations center (SOC) to see just the information that security teams need for damage control, keeping the noise in the background where it belongs. Table 1 compares automated and manual approaches.

**Table 1: Automated Versus Manual Approaches to Processing and Analysis**

	PROS	CONS
Automated	<ul style="list-style-type: none"> <li>» Fast</li> <li>» Predictable</li> <li>» Scalable</li> <li>» Able to process large amounts of information</li> </ul>	<ul style="list-style-type: none"> <li>» Must be properly configured</li> <li>» Cannot perform detailed analysis</li> <li>» Could miss critical information</li> </ul>
Manual	<ul style="list-style-type: none"> <li>» <i>Able to perform high-end analysis</i></li> <li>» Enables in-depth correlation</li> <li>» Facilitates ad hoc analysis and discovery</li> </ul>	<ul style="list-style-type: none"> <li>» Slow</li> <li>» Not scalable</li> <li>» Limited ability to process large amounts of information</li> </ul>

Source: illustration purposes only

Contextual visualization and filtering are two ways to provide security teams with useful intelligence that is both intuitive and actionable. The saying “a picture is worth a thousand words” is especially true when it comes to security; visual formats can simplify the processing of large amounts of information and threat alerts. Meanwhile, filtering ensures that the information the security team receives is of high value, with as little “noise” as possible. Every device on a network generates traffic, which can be overwhelming from an analysis perspective. Only through proper filtering can information of value be discovered.

Ultimately, the security team will benefit from a more efficient way to visualize data and metrics. Clear visualization and prioritization enables staff to better use valuable intelligence, which in turn leads them to make decisions in a timely manner. Although a comprehensive dashboard helps, multiple dashboards will hinder the security team’s ability to focus on the most valuable data. Ultimately, what matters most is the data feeding the dashboard.

Dashboards must combine automated and manual information; they cannot just provide data, but must facilitate cognitive reasoning and quick response. This in turn requires sophisticated visualization features that include high-end data aggregation, scrubbing and correlation. Such capabilities will enable incident responders to make proper decisions while offering them an intuitive visual console. A “single pane of glass” view of relevant data will enable security analysts to drill down and discover insights and patterns.

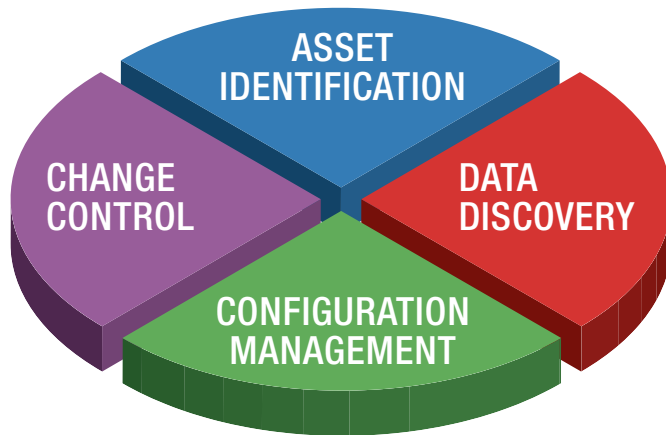
## A Smarter Approach to Security

The typical gated community reflects traditional approaches to designing and implementing IT security, with a defined perimeter and controlled access through it. Such communities may keep some “undesirables” out of the development, but they are vulnerable to anyone the guards recognize or who can jump over the fence. Likewise, an organization may have the best firewall available, but attackers who bypass it might find themselves in a network that is wide open because of a flat design, which makes it easy for them to access any information they want.

Instead, let’s look at the example of the residences within the gated community, each with an alarm system and locks of its own: there is a clear division of control between each home. If burglars hit one house, they do not find it easier to rob another, which means the amount of damage is managed and controlled. In a similar vein, networks within an organization should be highly segmented, thereby limiting the reach of any single machine. This way, if attackers compromise one system, they find it no easier to compromise others. Such segmentation also helps catch attacks early and control lateral movement into more sensitive areas in the network.

Although highly segmented networks are one of the more practical defensive tactics to follow, organizations must have

proper foundational items in place for any defensive effort to succeed. The four components of a solid cybersecurity foundation are shown in Figure 2.



**Figure 2: Components of a Sound Cybersecurity Foundation -**  
(Source: Author)

If an organization does not know what devices are on its network and how they are configured, is unable to manage change or does not know where its critical data is located, its security is doomed to fail. Although no solution on its own will stop attackers—they will always find a way into a system—a layered approach to security can provide a sound foundation.

Proper IT security is not about the quantity of information; it is about its quality. Large amounts of useless information can distract the security team, but prioritizing and focusing on high-quality information in an appropriate contextual perspective leads to useful intelligence. Consequently, contextual awareness leads to appropriate and timely decisions, which reduces dwell time and controls overall damage. Ultimately, a smarter approach to security requires a single visual interface with integrated metrics, one that visually allows an IT security team to quickly understand what is happening across the enterprise network, discover patterns, derive insights, and make effective and informed decisions.

For information to be truly useful, understanding the context of the information and what is actually occurring is essential. Security analysts can then prioritize and focus on the information that really matters, in turn enabling fast and decisive action; a clear visual interface is an essential tool for such work.

Effective security solutions must align with how attackers work and focus on controlling the amount of damage an organization will experience. If compromise is inevitable, then the next best approach is to contain and control threats so damage is limited. One way to approach this problem is to think of three pillars, as we see in Figure 3.



**Figure 3: Pillars of Cybersecurity Success -** (Source: Author)

These five points expand on the three pillars of success: detection, containment and control:

1. **Use Security tools with end-to-end visibility across the entire organization.** Correlating the universe of activity is essential to a full understanding of what is happening during an attack. Advanced threats are stealthy by design; if a security device is looking at only one aspect, it will most likely miss the attack. Point solutions alone are not effective. An all-encompassing view of the network with visibility into what is transpiring across the enterprise is necessary when attempting to detect and contain harmful activity.
2. **Beware information overload; too much visibility is almost as bad as too little.** Understanding the context of data is critical because attackers will try to mimic the patterns of normal user activity. Filtering out noise and focusing on the activities that really matter are the ways to a better understanding of contextual awareness, which requires correlating current activities with “known good” behavior to gain intelligence on what an adversary is doing and how. Gathering information on user activity helps provide proper contextual awareness of what is happening. For example, a user copying a 500MB file on a Saturday could be a problem, but understanding the context requires knowledge of the user’s other activities. Combining this intelligence with analytical capabilities provides specific insight into what a potential adversary is doing, which can be a basis for early detection, thereby containing a potential attack and controlling the overall damage to the organization.
3. **Get security solutions that perform real-time analysis.** Speed is essential when fighting an attack, and there’s just no substitute for real-time analysis. Of course, the success of such analysis requires the filtering of noise, so the security solution can work with just the information that is likely to detect an



attack. Data-driven intelligence is the key to quickly identifying, controlling and minimizing the damage caused by an attacker. This information needs to be presented to SOC analysts in a manner that enables them to make proper decisions. An intuitive visual interface based on cognitive research clearly displays what is happening, with proper context.

4. **Reduce the dwell time an adversary spends within a network.** The longer an organization is compromised, the greater the overall damage. Therefore, this is the most important point of all. Early detection and controlling the adversary are vital to reducing overall dwell time—and thereby reducing damage and related costs.
5. **Implement an in-depth defense.** Because adversaries often cannot directly break into the system they want to compromise, they will look for one that can be compromised and use it as a pivot point to go deeper into the network. Such lateral movement allows an adversary to cause more damage, which makes it even more surprising that many organizations focus on perimeter protection and completely miss internal activity. Effective security solutions must monitor the internal network, detect when systems are compromised, and be able to recognize the attacker's lateral movements.

## *Security is no longer just about setting up some devices at the network edge.*

In each case, the three pillars of successful defense represent challenges and opportunities for both evolving and mature security models.

Recognizing the speed and persistence with which adversaries break into systems, security is no longer just about setting up some devices at the network edge. It instead requires continuous monitoring with timely response – the most effective ways to minimize dwell time. Meeting this requirement has led many businesses to hire security analysts, or even establish a SOC. In the simplest sense, a SOC is responsible for monitoring and responding to the intelligence the organization's security devices generate. Many IT departments struggle when standing up their SOCs with establishing an appropriate focus for their work and defining the information that analysts can have; in other words, how to monitor. From an analyst's perspective, the most important part of monitoring—with or without a SOC—is to have an effective, properly designed visual interface. It should be easy to use and must be able to show

the analysts what is happening within the organization, providing situational awareness across the network and—given the growing use of cloud services—beyond. The visual interface must enable analysts to drill down into events, to better understand what is happening and verify the accuracy of the information to make effective and actionable decisions

The most important part of visual interface design is the data it measures and displays to the analyst. The single-pane-of-glass approach is critical if analysts are to discover abnormal activity in a timely fashion. Any visual interface must be properly integrated with other systems and provide accurate and clear information.

The problem is not that security teams need more metrics. Instead, they need the right metrics: data they can easily measure and act upon. Any monitoring interface must be dynamic, constantly tracking the adversary and providing information in a manner that leads to prompt and appropriate decisions. If the security dashboard is to clearly show deviations from normal activity, the metrics behind the dashboard must quantify the difference between normal and hostile activity. Security analysts need an intuitive, easy-to-use interface with visualization capabilities, so they can quickly see what is happening in their environment. The most critical metrics are those that are associated with data flows, both within a network and outbound. Monitoring suspicious connections—and the amount of information flowing over them—makes it possible to identify deviations caused by adversaries and take proper action.

Many people try to minimize the frequency of illness, but when they do get sick—because we all do—their goal is to minimize the impact of the illness. In cybersecurity, the goal is the same: minimizing the frequency in which an organization is compromised and, when a compromise occurs, responding swiftly to minimize the damage and exposure to the organization.

A key component of a successful security program is a SOC that can monitor and respond to attacks in a timely manner. An effective SOC relies on key metrics such as reducing dwell time and minimizing lateral movement, information that feeds into a dashboard and gives security analysts visibility into what is happening within an organization. By focusing on useful intelligence and implementing tools that enable real-time analytics, organizations can ensure analysts get the information they need, when they need it, to maintain proper security across the organization.

---

### ABOUT THE AUTHOR

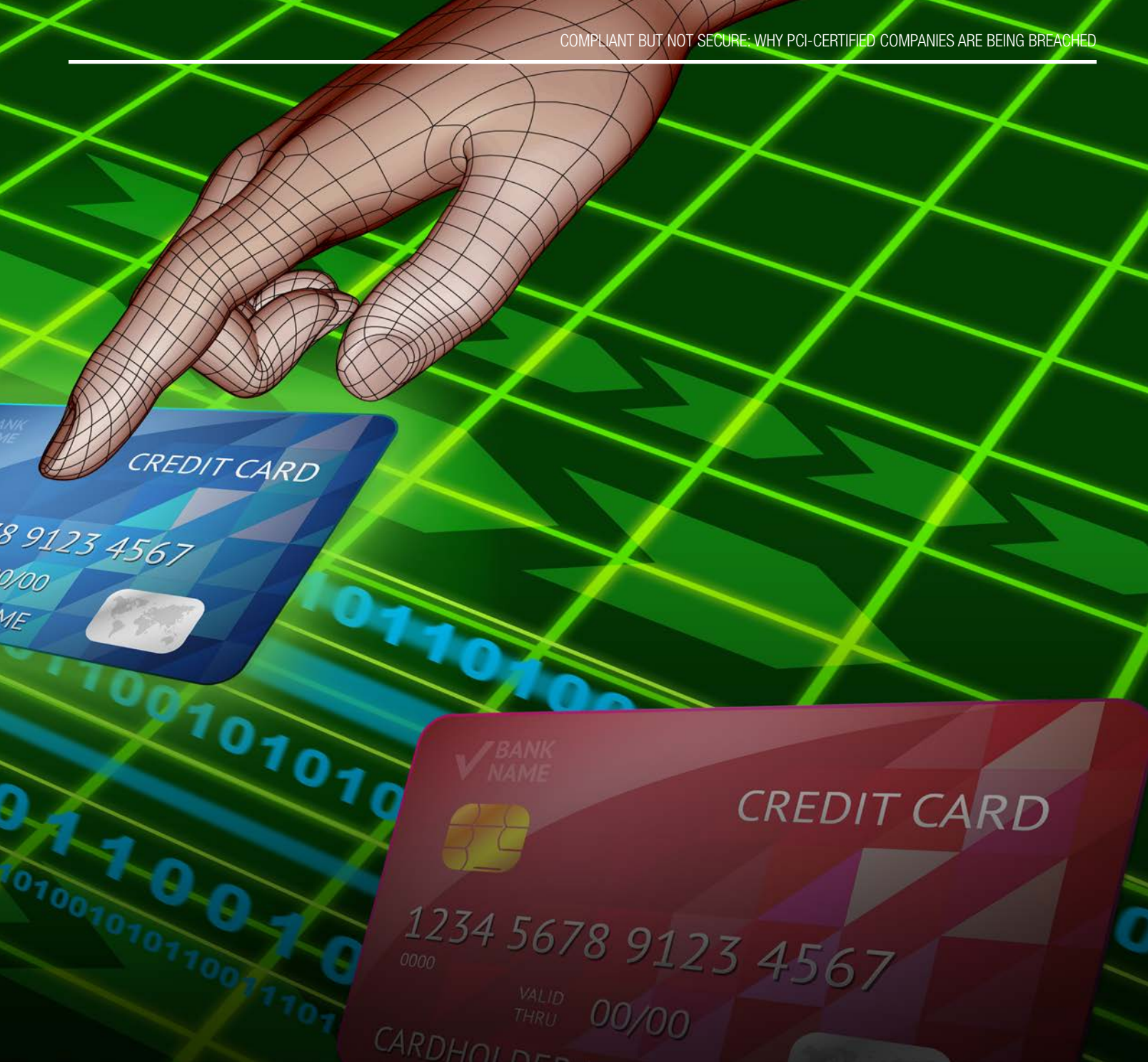
**Eric Cole, PhD**, is a SANS Faculty Fellow, course author and instructor who has served as CTO of McAfee and chief scientist at Lockheed Martin. He is credited on more than 20 patents, sits on several executive advisory boards and is a member of the president's Commission on Cyber Security. Eric's books include *Advanced Persistent Threat*, *Hackers Beware*, *Hiding in Plain Site*, *Network Security Bible* and *Insider Threat*. As founder of Secure Anchor Consulting, Eric puts his 20-plus years of hands-on security experience to work to help customers build dynamic defenses against advanced threats.



# COMPLIANT BUT NOT SECURE: Why PCI-Certified Companies Are Being Breached

By: Christian Moldes, Candidate, SANS Technology Institute, MS in Information Security Management

**The Payment Card Industry published the Data Security Standard 11 years ago; however, criminals are still breaching companies and getting access to cardholder data.** *The number of security breaches in the past two years has increased considerably, even among the companies for which assessors deemed compliant. In this paper, the author conducts a detailed analysis of why this is still occurring and proposes changes companies should adopt to avoid a security breach.*



## 1. Introduction

The Payment Card Industry Security Standards Council (PCI SSC) published the Data Security Standard (DSS) to provide a minimum set of required security controls to protect cardholder data 11 years ago (Search Security, 2013).

According to Verizon's PCI DSS Compliance Report, the number of organizations that are fully compliant at the time of interim assessment is growing rapidly each year. While the increase in organizations taking PCI DSS compliance seriously is important, there has been a rise in organizations' data breaches.

There are still many misconceptions about PCI DSS compliance and its role in providing a reasonable level of security. Some of these misconceptions have driven organizations to reallocate resources into preventive controls while disregarding detective controls. The resource allocation strategy may provide a low rate of successful implementation due to misaligned operational and strategic goals which could result in ineffective incident handling procedures and/or intrusion detection failures.

Organizations are being breached due to failure to implement the minimum set of security controls. This article will focus on the organizations which Qualified Security Assessors (QSAs) have deemed PCI DSS compliant.

## 2. Compliant but not Secure

One of the major misconceptions about PCI DSS compliance is PCI DSS-certified companies are secure or hacker-proof as vendors in the industry may carelessly advertise. In fact, according to Verizon's PCI DSS Compliance report, only 29 percent of companies are compliant a year after validation. This means that many businesses are checking the boxes for PCI DSS compliance off their list, or even just implementing compensating controls, and then forgetting about it until the next audit is due. In 2013, Target was certified PCI DSS compliant weeks before hackers installed malware on the retailer's network.

Others such as Heartland Payment Systems suffered a major breach even though assessors deemed their company compliant for six consecutive years.

Either the PCI DSS is an ineffective security standard for protecting cardholder data or the organization's implementation of PCI DSS is conceptually flawed in their approach. If PCI DSS does not guarantee security, what is the actual benefit of being compliant? Besides possibly providing some legal safe harbor, PCI-DSS compliance does not eliminate probability of payment data breaches.

PCI DSS includes security controls to deal with the most common risk scenarios and known attack vectors identified by the PCI SSC. Even though, PCI SSC continues to update the PCI DSS over the years, it's virtually impossible for PCI DSS to anticipate every possible attack scenario. While PCI Security Standards Council is constantly working to monitor threats and improve the industry's means of dealing with them, ultimately, it's each organization responsibility to provide credit card data security.

## 3. Why Are PCI-Certified Organizations Being Breached?

Verizon 2015 PCI Compliance Report states, "Of all the companies investigated by our forensics team over the last 10 years following a breach, not one was found to have been fully PCI DSS compliant at the time of the breach" (Verizon, 2015).

Based on the statistical information collected by Verizon, it is reasonable to assume that organizations may have met compliance standards; however, the security controls were not sustainable or resilient after the initial certification assessment. There are multiple probable reasons why this may occur, and we could easily group these reasons into two major categories: reasons attributable to the organization and reasons attributable to the QSAs that are certifying these organizations.

### 3.1. Reasons Attributable to the Organization

#### 3.1.1. Compliance program

**PCI DSS includes security controls to deal with the most common risk scenarios and known attack vectors identified by the PCI SSC.**

Some organizations falsely assume that PCI DSS compliance is merely passing their annual assessments and obtaining certifications. These organizations are employing compliance efforts into a singular event; however, failing to maintain compliance is part of the organization's continuous monitoring effort. It is not surprising that these organizations end up being primarily breached because of the deficiency of a mature compliance standard which address protection and security measures of cardholder data.

These types of organizations usually fail to:

- Identify all locations where cardholder data is stored and define their compliance scope accordingly
- Gain visibility and control of their payment channels that could result in unknown new cardholder data flows and repositories
- Monitor security controls and compliance periodically
- Provide adequate security awareness to all the organization's stakeholders to ensure PCI DSS required security controls are understood and applied to all the system components in scope
- Fill out compliance self-assessment questionnaires without validating security controls

For example, Sally Beauty's sysadmins were using a Microsoft Visual Basic scripts that contained their network administrator's username and password (Krebs, 2015a). This insecure practice is in clear violation of PCI DSS requirement 8.2.1 which demands all credentials be rendered unreadable during transmission and storage on all system components (PCI SSC, 2015).

#### 3.1.2. Unrealistic expectations

Organizations may have unrealistic expectations for their QSAs. For example, they expect their QSAs to:

- Understand the organization's business processes and applications better than the organization's staff
- Uncover all gaps and vulnerabilities
- Uncover all locations where the organization stores cardholder data

Even properly scoped assessments are limited by time and resources, and as such, in most cases QSAs can only review a sample of systems components. Making it impossible for a QSA to uncover all gaps and vulnerabilities. It is common for an organization that

has previously been marked PCI-compliant to remediate newly unidentified gaps during an assessment cycle.

An experienced QSA may be familiar with typical locations where organization store cardholder data and he or she may be able to find data stored at offsite data repositories. However, it will be a difficult task for a QSA to trace all locations where cardholder data is stored unless the organization is using an automated access control system.

For example, Forever 21 retails, after a security breach blamed their QSA for failing to uncover undisclosed files containing cardholder data (Schuman, 2008). Unless this QSA was hired to conduct a data discovery process, it is unreasonable to blame the QSA for these undisclosed data repositories.

### 3.1.3. Human error

As it is widely known in information security domain, humans are considered the weakest link in the security chain subsequently organizations should anticipate that people may inevitably fail. Employees may fail to apply a security patch, misconfigure a security setting, fail to follow security policies and procedures, or may become susceptible to phishing attacks. Regardless of the security controls in position, cyber criminals are effective with exploiting the irrational elements of human nature.

For example, a district manager that kept his credentials taped to a laptop may have contributed to Sally Beauty's security breach (Krebs, 2015a). This raises more questions about the effectiveness of Sally Beauty's security awareness program and its compliance with PCI DSS.

### 3.1.4. Focus on preventive controls only

Many organizations focus their entire operational efforts on security breach prevention while overlooking the importance of resource allocation to the cybersecurity incident response plan for detecting, analyzing, prioritizing, and handling incidents.

If cyber-criminals are successfully exploiting traditional measures of trust to gain a foothold on cardholders' data, then it is highly probable that organizations were unable to detect the intrusion, and/or regardless of the number of controls employed, the intrusion-detection capability were ineffective due to inadequate deployment of intrusion detection system (IDS) sensors.

According to the Verizon PCI Compliance Report, several breached organizations received alert notifications; however,

some organizations failed to thoroughly investigate these alert notifications when such traffic occurs.

For example, Target confirmed that the cyber-attack vectors against their retailer's point-of-sale (POS) systems triggered alarms and their information security team chose to ignore (Schwartz, 2014). Sally Beauty's Tripwire solution fired warnings when the intruders installed malware on their point of sale systems. Either the cyber security team was not monitoring the alerts or they ignored the alerts altogether (Krebs, 2015a). In a similar case, Secure Pay's web application security system triggered several alerts to block a specific external internet protocol (IP) address; nevertheless, cyber criminals were successful with exfiltrating the cardholders' data. (Krebs, 2014a).

## 3.2. Reasons Attributable to the QSAs

Qualified Security Assessor (QSA) companies are independent security organizations that have been qualified by the PCI Security Standards Council to validate an entity's adherence to PCI DSS. QSAs have certified non-compliant organizations that are nowhere near compliant which maybe highly attributable to the QSAs' inappropriate methodologies and/or attributable to unqualified consultants. Even in these cases, it is important to understand that the role of a QSA in a PCI DSS assessment is not to conduct a complete discovery of all non-compliant issues. The QSA's role is to provide an opinion on the compliance status of an organization based on the time allocated to interview the organization's staff, review a sampling of system components, and analyze evidence provided by the organization.

### 3.2.1. QSA methodology

QSA's methodology to conduct PCI DSS assessments may lead to certifying non-compliant organizations. Jennifer Bjorhus conducted several interviews with industry members who described the work conducted by the largest QSA company as "lax", not accurate, "glaring with errors", and poor quality (Bjorhus, 2014).

The following list illustrates cases where a poor methodology may lead to a flawed assessment:

- QSAs who rely mostly on their interviewees' statements to validate compliance
  - Some QSAs may accept their interviewee's statements at face value. They do not realize that sometimes interviewees are not necessarily the most authoritative person to speak on a subject or that they may just assume that security controls

**Even properly  
scoped assessments  
are limited by time  
and resources,  
and as such, in  
most cases QSAs  
can only review a  
sample of systems  
components.**

are in place, and that sometimes interviewees may rely on what their staff has told them without validating those assertions themselves.

- QSAs who solely rely on evidence provided by the organization
  - QSAs have to keep in mind that the organization may provide evidence of only selected system components that currently comply with PCI DSS. QSAs may miss the opportunity to uncover compliance deviations and issues if they only rely on screenshots or partial configuration reviews provided at the organization's discretion.
- QSAs who spend little to no time onsite
  - With little time to conduct an onsite review, it is very unlikely that the QSA would conduct a thorough analysis and detect not so evident gaps. News media identified at least one QSA company of performing assessments in a third or quarter of the time compared to other QSA companies (Grundvig, 2013).
- QSAs who don't take a representative sampling of system components
  - QSAs who do not take appropriate sampling sets may fail to identify gaps in the security management processes and patterns that contribute to security operations inconsistency.
- QSAs who are validating positives instead of negatives
  - QSAs who validate positives would focus on finding evidence of compliance. QSAs who validate negatives focus on finding evidence of non-compliance. It is very easy to validate positives, as a small sampling would be sufficient to believe that an organization is PCI DSS compliant. On the contrary, validating negatives requires spending more time to ensure no instances of non-compliance exists. This latter approach would obviously take more time and most QSAs do not usually practice it.

### 3.2.2. QSA individual expertise

The QSAs level of proficiency may also be a factor which may result in non-compliant organizations passing their assessments, for example:

- QSAs who fail to identify the right compliance scope for an organization
  - QSAs may incorrectly advise their clients to leave critical components out of the compliance scope. These components, if compromised, could be used by an attacker to gain access to the cardholder data environment.
- QSAs who are not experts on specific areas or technologies

- QSAs who are not experts on the technologies under review may fail to identify critical vulnerabilities or misconfigurations. An intruder may exploit these vulnerabilities to escalate privileges and gain access to cardholder data.
- QSAs who are not familiar with hacking techniques or attack vectors that hackers use to breach organizations
  - A QSA who is not familiar with hacking techniques or attack vectors may fail to identify how the lack of specific security controls could put the cardholder environment at risk. Robert Carr, Heartland's CEO, blamed his QSA for being unable to identify a common attack vector that criminals used against other companies (Brenner, 2009).

## 4. Improving PCI DSS Compliance and Security

Given multiple reasons why organizations may be compliant and yet not secure, organizations should strive to improve their compliance with the PCI Data Security Standard by taking holistic, or a tiered approach to improve the organizations' security posture.

### 4.1.1. Develop a mature compliance program

Organizations should develop a mature compliance program by conducting the following tasks:

1. Designate an individual or group to manage and monitor PCI DSS compliance and empower them to have influence across the organization.
2. Conduct a data discovery process regularly to identify and maintain an inventory of data repositories and system components in scope. Define your PCI DSS scope based on this inventory.
3. Automate PCI DSS compliance to have a clear visibility of the compliance status of the organization at all times. Organizations can achieve this task by using GRC tools such as IBM OpenPages, RSA Archer or similar tools.
4. Provide appropriate security awareness training to ensure all stakeholders understand the need of PCI DSS compliance. This training has to be tailored to the specific needs of each organizational group.
5. Follow PCI SSC's best practices for implementing PCI DSS into business-as-usual processes.

### 4.1.2. Select the right QSA

Organizations should understand that PCI Compliance is the organization's responsibility, not the QSA's responsibility. Though, not having well-qualified QSAs may encumber his or her ability to interpret 'state of the art' security and ensure that controls are commensurate with risk.

Price should not be the only factor to take into consideration when selecting a QSA. Consider the QSA methodology, assessment process, and internal training practices as well. Keep in mind that small consulting companies may lack the corporate knowledge of large QSA companies. There is strength in numbers; therefore, large QSA companies may be more profitable based on their global talent pool through various expertise, diversity of opinions, and insight of multiple industries.

Interview your QSA consultant before committing to an assessment. Select your QSA consultants based on their expertise and knowledge of your industry, technologies in use, and information security. Keep in mind that QSA consultants cannot be experts on everything but at least some exposure to the business processes and technologies used by your organization is very important. A QSA consultant with some experience in penetration testing or computer forensics is highly desirable. These individuals would be able to identify vulnerabilities easily based on their insight of past security breaches and hacking techniques, and your organization would obtain the most value out of each assessment cycle.

It is important to rotate QSA consultants at least every couple of years. Your organization may benefit from having different perspectives, expertise, audit skills, and vast approaches to the PCI DSS assessment.

#### *4.1.3. Strengthen your monitoring and investigation capabilities*

In the era where Advanced Persistent Threats (APTs) are more prevalent, organizations are realizing the dangers that could lurk around the virtual corner. Cyber hackers may spend as much time as needed to perform reconnaissance, research of the organization and technologies in use, to include obtaining information about the organization's security controls in place.

Researchers found that the malware used in the Target's security breach was custom-tailored for the intrusion which was carefully written to avoid detection by standard antivirus software on the market (Krebs, 2014b).

Organizations have to allocate more resources to strengthen their monitoring and investigation capabilities. Organizations should document their assets plus locations, network dataflow diagrams, identify potential threat vectors and the attack surfaces within them. The staff assigned to monitoring activities should support the cyber security initiatives through both predictive and reactive analysis, articulating emerging trends, perform network traffic analysis utilizing raw packet data, net flow, IDS,

and custom sensor output as it pertains to the cyber security of communications networks.

Organizations with limited resources should at least adopt risk-based monitoring process. For example, system components could be classified according to criticality:

- a. Group 1: All system components that store cardholder data
- b. Group 2: All system components that process and transmit cardholder data but which do not store it even temporarily.
- c. Group 3: All system components that provide security and authentication services
- d. Group 4: All system components that provide access to the cardholder data environment
- e. Group 5: All system components that are facing external networks such as the Internet, partners' networks, or wireless networks.
- f. Group 6: Any other components in scope not included in previous groups.

Ideally, organizations should monitor and investigate all the security events and alerts; however, assuming that resources are limited, organizations could use the following strategy to monitor and investigate activities:

- a. 50% of monitoring time assigned to group 1 and 2. The organization should investigate all the security events and alerts in this group.
- b. 35% of monitoring time assigned to groups 3, 4 and 5. The organization should investigate all the critical events in this group and remaining events only if there is time left.
- c. 15% of monitoring time assigned to group 6. The organization should sample security events and alerts in this group for additional research and investigation, and pick different types of events each day.

Organizations should learn from their own and other organizations' mistakes. Special attention should be paid to attack vectors successfully used during previous penetration tests and for the techniques and attack vectors used by criminals to breach other organizations.

## **5. Conclusion**

There are multiple links between PCI DSS compliance and an organization's ability to defend itself against potential cyber

***In the era where  
Advanced Persistent  
Threats (APTs) are  
more prevalent,  
organizations  
are realizing the  
dangers that could  
lurk around the  
virtual corner.***

breaches; however, still many organizations are failing to maintain compliance. Although it is great to see PCI compliance increasing over the years, nevertheless the fact remains that organizations whether large or small are still not meeting PCI DSS standards. These PCI program issues may be attributable to the organizations failing to comply with PCI Data Security Standard (DSS) or Payment Card Industry Qualified Security Assessor (QSA) companies failing to identify security issues during the initial assessment. These are serious concerns, because cyber criminals are staying ahead of the curve, and with increasing connectivity through technology, attacks may originate from anywhere in the world. We face a perilous cyber world that threatens organization's ability to safeguard both data in transit and at rest therefore maintaining PCI compliance should be employed as the defense against manner of nefarious cyber activities. Organizations must continue to focus on the goal of safeguarding customer data, not just pass the PCI DSS assessment. Consumers are counting on organizations to secure data in transit while providing appropriate level of vulnerability management and overall risk management.

## REFERENCES

- [1] Bjorhus, J. (2014). "Clean Reviews Preceded Target's Data Breach, and Others". Retrieved August 15, 2015 from [www.govtech.com](http://www.govtech.com) website: <http://www.govtech.com/security/Clean-Reviews-Preceded-Targets-Data-Breach-and-Others.html>
- [2] Brenner, B. (2009). "Heartland CEO on Data Breach: QSAs Let Us Down". Retrieved August 15, 2015 from [www.csoonline.com](http://www.csoonline.com) website: <http://www.csoonline.com/article/2124260/privacy/heartland-ceo-on-data-breach--qsas-let-us-down.html>
- [3] Grundvig, J. (2013). "Changing Your Password Won't Change Anything - You Will Still be Hacked". Retrieved August 15, 2015 from [www.huffingtonpost.com](http://www.huffingtonpost.com) website: [http://www.huffingtonpost.com/james-grundvig/changing-your-password-wo\\_b\\_4414149.html](http://www.huffingtonpost.com/james-grundvig/changing-your-password-wo_b_4414149.html)
- [4] Krebs, B. (2014a). "Thieves Jam Up Smucker's, Card Processor". Retrieved August 15, 2015 from [krebsonsecurity.com](http://krebsonsecurity.com) website: <http://krebsonsecurity.com/2014/03/thieves-jam-up-smuckers-card-processor/>
- [5] Krebs, B. (2014b). "A Closer Look at the Target Malware, Part II". Retrieved August 15, 2015 from [krebsonsecurity.com](http://krebsonsecurity.com) website: <http://krebsonsecurity.com/2014/01/a-closer-look-at-the-target-malware-part-ii/>
- [6] Krebs, B. (2015a). "Deconstructing the 2014 Sally Beauty Breach". Retrieved August 15, 2015 from [krebsonsecurity.com](http://krebsonsecurity.com) website: <http://krebsonsecurity.com/2015/05/deconstructing-the-2014-sally-beauty-breach/>
- [7] PCI SSC (2015). "PCI DSS v.3.1". Retrieved August 15, 2015 from [www.pcisecuritystandards.org](http://www.pcisecuritystandards.org) website: [https://www.pcisecuritystandards.org/documents/PCI\\_DSS\\_v3-1.pdf](https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-1.pdf)
- [8] Schuman, E. (2008). "Breach Update: Forever 21 Stored 5-Year-Old Transaction Data". Retrieved August 15, 2015 from [archives.thecontentfirm.com](http://archives.thecontentfirm.com) website: <http://archives.thecontentfirm.com/securityfraud/breach-update-forever-21-stored-5-year-old-transaction-data/>
- [9] Schwartz, M. (2014). "Target Ignored Data Breach Alarms". Retrieved August 15, 2015 from [www.darkreading.com](http://www.darkreading.com) website: <http://www.darkreading.com/attacks-and-breaches/target-ignored-data-breach-alarms/d/d-id/1127712>
- [10] Search Security. (2013). "The history of the PCI DSS standard: A visual timeline". Retrieved August 15, 2015 from [searchsecurity.techtarget.com](http://searchsecurity.techtarget.com) website: <http://searchsecurity.techtarget.com/feature/The-history-of-the-PCI-DSS-standard-A-visual-timeline>
- [11] Verizon (2015) "Verizon 2015 PCI Compliance Report". Retrieved September 3, 2015 from [www.verizonenterprise.com](http://www.verizonenterprise.com) website: <http://www.verizonenterprise.com/pcireport/2015/>

## ABOUT THE AUTHOR

**Christian Moldes** is currently working for IBM as the PCI Services Technical Manager for North America but supporting PCI DSS related projects across the world. In this role, he acts as the global subject-matter expert at IBM, acts as an escalation point for clients and consultants, and delivers PCI DSS advisory and assessment services to internal and external customers. He is also the primary contact between IBM and the PCI Security Standards Council.

With more than 24 years in IT auditing and security consulting, Mr. Moldes has delivered information security services in a diversity of areas and industries. He holds a BS in Computer Science and is currently enrolled in the Master degree program in Information Security Management at SANS Institute.

He has obtained the following certifications: CISM, CISSP, MCSE: Security, CISA, CCNA, PCI QSA, GIAC GCIH / GLEG / GSSP-NET, GWEB, GCPM, GSCC, ISMS Lead Auditor (ISO 27001:2005) and others vendor related.

He loves traveling and has delivered services for organizations in North America, Latin America and Caribe, Europe, Eastern Europe, Africa, and Asia.

He loves meeting people from different cultures and speaks English, Spanish, Portuguese, elementary French, and is currently learning Chinese (Mandarin).





# Need Specialized Technical Support with Easy Contract Terms?

## Core Analysis Task (CAT) Program

*A Pre-Awarded, Pre-Competed Contract Vehicle.*

CSIAC provides Subject Matter Expert (SME) support on an as-needed basis to quickly address technical requirements with minimal contracting effort. CSIAC provides such solutions via the utilization of our Core Analysis Task (CAT) service/capability. CSIAC is a competitively awarded contract with Indefinite Delivery/Indefinite Quantity (ID/IQ) provisions that allow us to rapidly respond to our users' most important needs and requirements. Custom solutions are delivered by executing user-defined and funded CAT projects without the need for further competition.

Through the CAT program, CSIAC is a pre-competited contracting vehicle, enabling the DoD and other agencies to obtain technical support for specific projects/programs that fall within one of the CSIAC technology areas. As with any inquiry, the first four hours are free. If the scope requires a CAT, CSIAC will assist with the development of a Performance of Work Statement (PWS) to be approved by the Contracting Officer's Representative (COR).

### Key Advantages of working with CSIAC:

#### **Expansive Technical Domain**

The CSIAC's broad technical scope provides numerous pre-qualified resources for potential projects, and is especially valuable for today's information system challenges that frequently cross multiple domains.

#### **Comprehensive STI Repositories**

As a consolidation of three predecessor Information Analysis Centers (IACs), CSIAC has a wealth of expertise, data and information to support the successful completion of CATs.

#### **Expansive Subject Matter Expert Network**

CSIAC is able to leverage reach-back support from its expansive SME Network, including technical experts from the CSIAC staff, team members, or the greater community, to complete CATs.

#### **Minimal Start-Work Delay**

Not only does CSIAC provide DoD and other government agencies with a contract vehicle, but as a pre-competited single award CPFF IDIQ, work can begin in just a matter of weeks.

#### **Apply the Latest Research Findings**

CSIAC draws from the most recent studies performed by agencies across the DoD, leveraging the STI holdings of the Defense Technical Information Center (DTIC). The results of all CSIAC CATs and other DoD-funded efforts are collected and stored in DTIC's STI repository to support future efforts by the CSIAC and others.

## How To Get Started

If you have a need for CSIAC technical support, the first step is to contact us. All Technical Inquiries are free to the customer for up to four hours of service. If the scope of the support is more extensive and requires a CAT, CSIAC will assist with the development and submission of the task description and related contract documents. CATs may be awarded as either Cost Plus Fixed Fee (CPFF) or Firm Fixed Price (FFP) delivery orders.

Inquiries may be submitted by email to [info@csiac.org](mailto:info@csiac.org), or by phone at **1-800-214-7921**.

**Please visit our website for more information:**

<https://www.csiac.org/services/core-analysis-task-cat-program/>

## Who We Are

The Cyber Security Information Systems Information Analysis Center (CSIAC) is the DoD's Center of Excellence in Cyber Security and Information Systems, covering the following technical domains:

- Cybersecurity
- Software Engineering
- Modeling and Simulation
- Knowledge Management/  
Information Sharing

CSIAC is chartered to leverage best practices and expertise from government, industry, and academia to solve the most challenging scientific and technical problems. The Center specializes in the collection, analysis, synthesis, and dissemination of Scientific and Technical Information (STI) to produce solutions in support of the defense community.

## Our Team

Quanterion Solutions Incorporated is the prime contractor responsible for operating the CSIAC. In addition to Quanterion, customers also have access to the other members of the CSIAC team which include leading technology corporations as well as prestigious academic institutions that perform cutting edge research activities to expand our knowledge base.



Cyber Security & Information Systems  
Information Analysis Center

266 Genesee Street  
Utica, NY 13502

1-800-214-7921  
<https://www.csiac.org>



# ACCESSING THE INACCESSIBLE: INCIDENT INVESTIGATION IN A WORLD OF EMBEDDED DEVICES

By: Eric Jodoin, Graduate, SANS Technology Institute, MS in Information Security Engineering

**There are currently an estimated 4.9 billion embedded systems distributed worldwide. By 2020, that number is expected to have grown to 25 billion.**

*Embedded systems can be found virtually everywhere, ranging from consumer products such as Smart TVs, Blu ray players, fridges, thermostats, smart phones, and many more household devices. They are also ubiquitous in businesses where they are found in alarm systems, climate control systems, and most networking equipment such as routers, managed switches, IP cameras, multi-function printers, etc. Unfortunately, recent events have taught us these devices can also be vulnerable to malware and hackers. Therefore, it is highly likely that one of these devices may become a key source of evidence in an incident investigation. This paper introduces the reader to embedded systems technology. Using a Blu ray player embedded system as an example, it demonstrates the process to connect to and then access data through the serial console to collect evidence from an embedded system nonvolatile memory.*



## 1. Introduction

In a world where the Internet of Things is becoming a thing, embedded devices have become ubiquitous. In fact, nearly all classes of electronic devices are becoming embedded systems. According to a recent Gartner analysis, there are currently 4.9 billion embedded systems in use worldwide, and the number is expected to grow to 25 billion by 2020 (Gartner, 2014). Embedded devices can be found in a growing number of businesses with industrial grade appliances including routers, switches, IP cameras, alarm systems, lighting and climate controls, multi-function printers, and a rising number of consumer electronic goods such as smart TVs, Blu-ray players, fridges, thermostats, smart phones, etc. Even modern Supervisory Control and Data Acquisition (SCADA) systems are considered embedded systems. In fact, there is an increasing demand for embedded devices with greater computing power, better connectivity, and broader functionality while keeping implementation costs as low as possible. As a result, the vast majority of manufacturers have adopted some form or other of embedded Linux OS because of its relatively low cost, broad community support, and compatibility with an extensive range of hardware.

Most of these embedded devices come with custom-made user interfaces meant to simplify and constrain the end-user/administrator interaction to very specific actions and displays. This is helpful for preventing most accidental or intentional acts that could render the device irrevocably inoperable. But, there is a lot more going on under the hood. Logs are being generated, data is being saved to flash memory, changes are made to the device's configuration files, and more—a lot of which remains inaccessible using the custom-made interface.

In addition, embedded device security has been improving over these past few years. Not so long ago, it was rare to find an updated embedded system. This resulted in numerous avenues of attack opening up whenever a new vulnerability with Linux was discovered. Now, many devices update themselves automatically while others consistently remind the end user to authorize the update. However, there always remains a window of opportunity open between the time a vulnerability is discovered and the time a patch is engineered into an embedded system then deployed (Barry & Crowley, 2012).

Case in point, in January 2014, a security research company by the name of Proofpoint<sup>1</sup> uncovered a botnet

composed of more than 100,000 everyday consumer devices, such as home-networking routers, connected multi-media centers, televisions and at least one refrigerator, that had been compromised and used as a platform to distribute phishing and SPAM emails (British Broadcasting Corporation (BBC) 2014).

Therefore, the embedded devices running within an environment usually are limited in functionality and more effort is needed to access and are not infallible. What if the next embedded device malware does something more nefarious than sending SPAM and an analysis of the device is required to assess the impact to the organization? Perhaps there are clues that can be found on the embedded device? Unfortunately, the user interface does not provide the necessary means to easily access these hidden parts. But in most cases, there is another way of getting to this data.

## 2. Accessing an embedded device via the serial port

### 2.1. Embedded system primer

Embedded systems are custom purpose computers that are typically designed for one type of application. Design trade-offs are made to accommodate size, power, and application requirements. As a result, embedded computer systems are different, both from one another and from general-purpose computers by the virtue of their design trade-off and the constraints they embody (Barry & Crowley, 2012). But, regardless of these trade-offs, all systems share a common set of core features and capabilities whose understanding unlocks the ability to manipulate and access them in ways that are far beyond the means provided by the device's standard user interface.

#### 2.1.1. Embedded System anatomy

Unlike a traditional PC that comes equipped with as much functionality and as many expansion ports as possible, an embedded system usually comes with the bare minimum required to carry out



its assigned functions. Some embedded systems have a powerful standalone CPU and supporting chipsets to accommodate diverse subcomponents while others use a System-on-Chip (SOC) approach where all subcomponents are directly integrated with the CPU. Regardless of the approach, CPU/chipset combo or SOC, an embedded system, can be expected to be equipped with a number of standard subcomponents. This includes volatile memory, such as SDRAM, where the OS and programs will be run from, non-volatile memory, such as NAND Flash memory for data storage, and input/output (IO) controllers to allow the system to interact with its environment. It is virtually impossible nowadays to find an embedded system without serial, USB, and networking (Ethernet and/or Wifi) controllers included either on the board or directly integrated into the SOC processor.

2.1.2. Universal Asynchronous Receiver/Transmitter (UART) controller

Of particular interest for this paper is the serial communication controller, also referred to as the Universal Asynchronous Receiver/Transmitter (UART) controller. This is one of the most basic components that can be found on an embedded system and uses only 3 wires: Transmit (TX), Receive (RX), and Ground (GND). It is used extensively by system engineers during the development and testing phases. By connecting to this controller, it becomes possible to read hardware boot sequence messages, interact with the bootloader, and gain console access once the embedded Linux OS is loaded.

**Design trade-offs are made to accommodate size, power, and application requirements.**

Using one of several communication standards, the UART controller is responsible for all the tasks, timing, parity checking, etc. needed for the serial communication to succeed. The most commonly used standard is RS-232, which is also conveniently available on most PCs and laptops. If the incident handler's computer is not equipped with a serial port, a USB-to-Serial adapter can be easily procured online and configured to work with most Windows and/or Linux platforms as described later in this paper.

Each transmission flowing through the serial controller must obey a set speed limit and adhere to specific communication parameters that consists of a start bit, data bits, an optional parity bit and stop bits. To successfully communicate, both the embedded system and the incident handler's computer bit speed, data length, parity, and stop bits must be set to the same values. Much like a modem, there are many different transmission speeds supported: 300, 600,

1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, and 115200 bits per second (bps). However, the most commonly used speeds are 9600 bps, 38400 bps, and 115200 bps. The other most commonly used settings are no parity, 8 data bits, and one stop bit usually written as 8N1 (Barry & Crowley, 2012).

On the incident handler's computer, a terminal console application is used to communicate through the computer serial port with the embedded system. Because the embedded system will have its serial communication settings pre-configured, the incident

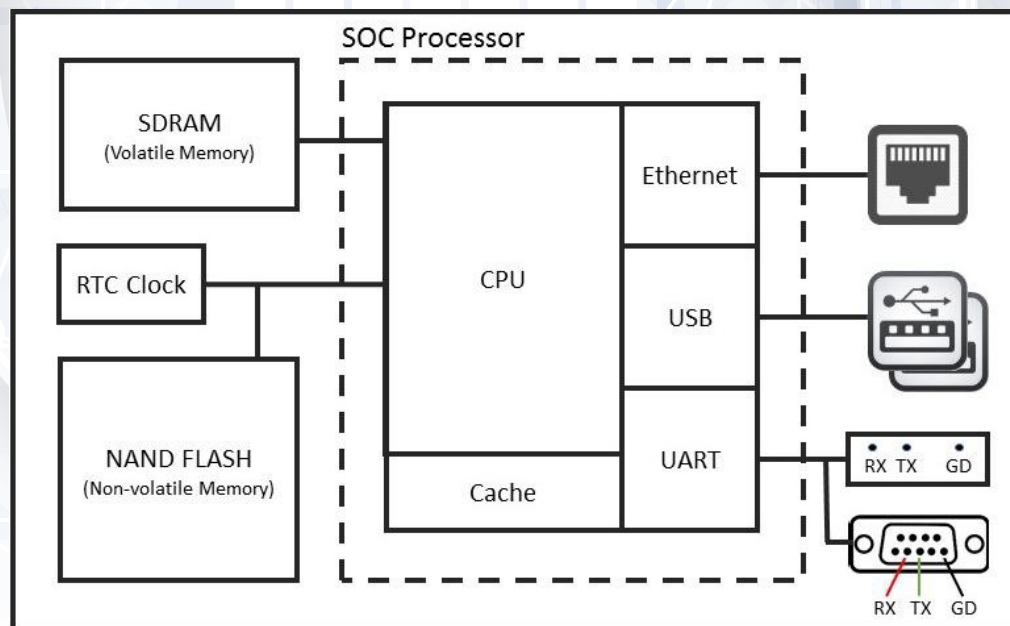


Figure 1: Simple Embedded System Using SOC Processor - (Source: Author)

handler's computer terminal console will need to be configured to match the embedded system settings in order to communicate successfully. The easiest way to find out if the embedded system uses the RS-232 standard and what communication settings are pre-configured is to lookup the embedded device specifications online. If it is not clearly stated on the device specification documentation, then looking up the specifications of the SOC processor or chipset used by the embedded device might yield the required information. As a last resort, it is possible to try various configuration settings until the terminal console outputs meaningful text. Configuring the communication protocol through the terminal software is relatively simple and will be explained later in section 2.4.

Finally, the RS-232 specification allows for voltages ranging between 5 and 25 volts (Electronic Industries Alliance (EIA) Standards, 1969). 5 volts is the standard voltage found on computer serial ports. It is also the standard voltage found on the serial connectors of most computing and networking gear equipped with an external serial port. However, some of the more recent embedded systems employ a modified UART controller operating at 3.3 volts. Although the RS-232 specification explicitly requires the UART controller to be able to work at voltages as high as 25 volts, discussion forums are littered with posts of people having "fried" their embedded device UART controller running at 3.3 volts after having connected it directly to a PC serial port running at 5 volts. The obvious solution to avoid such problems is for the incident handler to use a computer equipped with a serial connector with matching voltage. Methods to measure an embedded device's UART voltage is discussed later in section 2.3.2 while USB-to-Serial adapters able to supply different voltages are shown in section 2.2.1.

### 2.1.3. Embedded system boot process

When power is applied to an embedded device, the hardware begins its initialization sequence starting with the CPU. The CPU fetches the hardware initialization code, called preloader (or BIOS), from a specific flash storage chip. The preloader is software stored in flash memory that provides a consistent set of OS-agnostic software interfaces that abstract the underlying details of the hardware (Rothman & Zimmer, 2013). The preloader is also responsible for initializing the remainder of the embedded device hardware including volatile memory, also known as RAM, and IO components including the UART, PCI Bus, USB, and SATA controllers. Because of its simplicity, UART is most often the first communication port used to communicate debugging information from an embedded device. Finally, the last task of the preloader is to initialize the system storage, identify the boot device, and transfer control to the next agent in the boot process: either the bootloader or the operating system directly (Barry & Crowley, 2012).

***CPU fetches  
the hardware  
initialization code,  
called preloader  
(or BIOS).***

The concept of bootloaders is universal to virtually all operating systems, whether a full-fledged PC or the smallest embedded device (Waqas, 2010). The bootloader's main responsibility is getting the operating system from wherever it is stored, loading it into RAM and launching it. Most bootloaders can support booting from multiple storage locations and some even support loading OSes from the network or external storage temporarily attached to the embedded device. The bootloader will generally output boot diagnostic data to the serial port. It may even be possible for a short period of time to interrupt or at least interact with the bootloader through a terminal console connected to the device serial port. This can be particularly useful to:

- a. modify the Linux boot arguments;
- b. instruct the bootloader to load a different OS;
- c. save the firmware, embedded OS, or file partitions over the network or to a portable device; or,
- d. replace the existing embedded OS with new one from network or removable media.

However, functionality greatly differs amongst the various bootloaders available and with the exception of modifying the Linux boot arguments, interaction with the bootloader is beyond the scope of this paper.

The Unified Extensible Firmware Interface (UEFI) is a more sophisticated, 2<sup>nd</sup> generation preloader/BIOS that has become the de-facto firmware for most PCs. It is also starting to make inroads into embedded systems built around Intel SOCs. UEFI brings numerous improvements over the older technology such as only loading an OS with signed code. Bootloader programs that work with preloader/BIOS firmware are incompatible with UEFI. That said, most bootloaders have

been ported and now support the more advanced features provided by UEFI. However, for the purpose of this paper, preloader, BIOS, and UEFI can be used interchangeably.

In an effort to reduce costs, heat, and power use, most embedded systems are designed around an ARM SOC instead of using Intel technologies. Until very recently, there was no equivalent to the BIOS for the ARM processor. Each instance of ARM Linux kernel had to be hard coded for the hardware it was meant to execute on (Rothman & Zimmer, 2013). Hence, a number of ARM based embedded systems will be hardcoded to forego the bootloader and will be able to immediately load the OS, while other systems will use a very basic bootloader only responsible for loading and transferring control to the OS. In all cases, the OS kernel will take on the task of initializing the hardware. Newer designs based on the Open Firmware standard incorporate a more advanced preloader and bootloader architecture for the ARM

processor that is increasingly capable of abstracting the underlying hardware. From the incident handler's point of view, being able to determine if a bootloader is present or if the system boots Linux directly is important to assess possible lines of investigations.

## 2.2. Tools required

### 2.2.1. Hardware

The list of hardware components required is as follows:

- Incident handler's computer or laptop;
- USB-to-Serial adapters capable of operating at 3.3v or 5v as required;
- Serial cables and/or jumper wires;
- Solderless header pins (optional);
- Magnifying glass or magnifying glass app on smart phone (optional);
- Soldering iron and solder (Optional); and,
- Digital multimeter.

The incident handler's computer used for accessing the embedded device must be equipped with a USB port. A serial connector would also be convenient to connect to embedded devices with 5 volts serial interfaces but that can also be easily substituted with the appropriate USB-to-Serial adapter. The incident handler must also have root access to be able to install the necessary drivers and applications. For this paper, a VMWare version of Kali was used, allowing for full control of the OS and the ability to plug in the necessary USB-to-Serial adapters.

As explained in section 2.1.2, not all embedded systems have serial connectors operating on 5 volts. Furthermore, if the embedded system offers a 3.3 volts serial connection, instead of using the incident handler's PC serial connector, a USB-to-Serial adapter based on the PL2303HX<sup>2</sup> or similar chip will be necessary to synchronize the voltage with the embedded system. USB-to-Serial adapters supplying various voltage levels, including the more common 3.3 volts and 5 volts varieties, can be found online for less than \$10 USD each. For the sake of brevity, all examples discussed in this paper will assume the use of a 3.3 volts USB-to-Serial adapter equipped with female connectors as shown on the left in figure 2 below.



Figure 2: USB-to-Serial Adapters - (Source: Author)

Cables are obviously required to connect the computer and the embedded device together. If the device has an external serial port, then a standard null-modem DB9 serial cable or an RJ-45 to DB9 null-modem serial cable will be required. Null-modem means that the cable will connect the transmit line at one end of the connection to the receive line at the other end and vice versa. However, it is far more likely there will be no connectors present on the embedded device. In this case, it will be necessary to open the embedded device enclosure and scrutinize the Printed Circuit Board (PCB) with a magnifying glass for the transmit (TX), receive (RX), and ground (GND) port headers.

Then, a set of either male-male, female-female, or male-female jumper wires will be required to connect the incident handler's computer directly to the port headers on the PCB. It is even possible for the circuit board to have holes for the serial connector but no header pins soldered on. In this particular case, solderless header pins can be used to act as an impromptu connector. Finally, if all that is available on the PCB are contact pads, then a soldering iron, solder, and a pair of steady hands will be required to affix jumper wires and run them back to the USB-to-Serial adapter. The process to find the RX/TX/GND port headers is explained in detail in section 2.3.



Figure 3: Cables and Wires - (Source: Author)

Finally, a digital multimeter will be necessary to measure the voltage on the embedded system serial connector to prevent damage to the UART comptroller as described in section 2.1.2. The multimeter is also necessary for finding the UART TX, RX and ground (GND) port headers on any embedded system PCB lacking a serial connector. At last, the multimeter will be used to conduct continuity tests to ensure all cables and connections are adequate. A suitable digital multimeter can be found online for less than \$10 USD.

### 2.2.2. Software

This paper was developed using Linux Kali 3.18.0 64-bit. Linux was selected over Microsoft Windows because the PL2303HX drivers came pre-installed and they are loaded automatically when the adapter in plugged-in. minicom is a text-based control

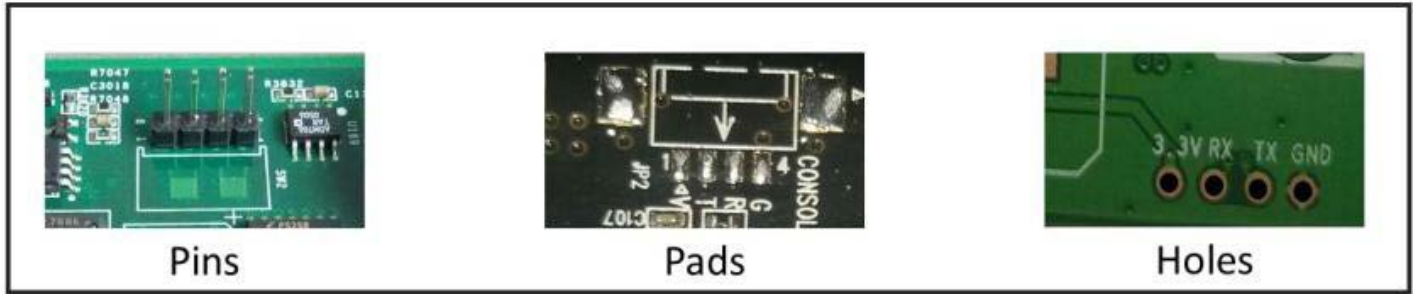


Figure 4: Serial Port Headers - (Source: Author)

and terminal emulation program for Unix-like operating systems capable of serial communication (Lackorzynski & Godisch, n.d.). It is free and can be found on most Linux distributions. minicom interfaces directly with the serial ports available on the computer and enables 2-way communications with the embedded device through the serial cable.

Minicom is relatively easy to operate. The command menu can be summoned at any time using the sequence **CTRL-A** followed by **Z**. There are numerous options available. However, the most important are the ability to select the desired serial port and the ability to change the communication parameters (speed, parity, stop bit(s), etc).

Configuration and utilization of minicom will be demonstrated later in section 2.4.

### 2.3. Connecting the serial port to the incident handler's computer

The procedure to locate and connect to an embedded device serial port is quite simple. However, precautions must be taken to eliminate the risks of causing irreparable damage to the device or the USB-to-Serial adapter. Furthermore, it is important to take additional precautions, such as using an anti-static mat and bracelet, when manipulating a PCB to prevent damage caused by accidental static electricity discharges.

Under the best of circumstances, the embedded device will be equipped with an external serial connector, and the incident handler will only have to run a null-modem cable between the computer and the device. This is often the case for networking equipment such as enterprise grade routers and switches. However, for other types of embedded devices, it is more likely that the incident handler will need to open the device and search the PCB for the port headers.

#### 2.3.1. Finding the port headers

In most cases, the serial port headers are labeled and easy to recognize. They will take one of three forms: A set of pins, a set of

pads or a set of holes as shown in figure 4 below. It is most often composed of four individual port headers and the vast majority of the time, they are arranged in a single row close to each other (Ganssle, et al., 2008). The first header is the ground (GND). The next header is the Transmit (TX) port. The last header is the Receive (RX) port. Note that a fourth header is also usually present. It is the Vcc header and it provides a steady voltage, usually 3.3 or 5 volts. While useful to help identify a set of serial port headers, it will otherwise not be required for the purpose of this paper.

The easiest method to find the serial header ports is to do a quick google search for the embedded device model number and the words "UART" and "Serial". If the device is widely popular, such as a Kindle Fire<sup>3</sup>, a Raspberry Pi<sup>4</sup>, or a NEST thermostat<sup>5</sup>, it is very likely that someone else has done the research, located the headers; and most importantly, documented the communication parameters and voltage required.

The next best approach is to open the embedded device and conduct a visual search. It is very likely that the words "GND, TX, RX, and 3.3 (or 5v or VCC)" will be printed on the PCB right next or very near their respective headers as shown in figure 4 above.

The location, size, and shape of the header ports will vary from system to system. As a result, some critical thinking and testing may be required. For example, some systems only have the TX and RX headers labeled. In such cases, the USB-to-Serial ground wire can be connected to any shielding or other parts of the PCB identifiable as a ground. And, in some rare instances, the UART port may simply not be labeled at all. Finding the port headers under this circumstance is beyond the scope of this paper. However, help is available online for the interested reader.

#### 2.3.2 Confirming the serial port headers

Regardless of the method used to locate the serial port headers, it is strongly recommended to verify each port header individually using a multimeter to avoid causing irreparable damage to the embedded device or the serial adapter on the incident handler's computer. The most important characteristic to verify is the voltage between the ground port header and each of the other port headers, none of which should exceed by more than 10%

the voltage rated for the serial connector used on the incident handler's computer.

The ground pin can be easily confirmed by carrying out a continuity test. Simply set the multimeter to the lowest Homs ( $\Omega$ ) setting available. The multimeter readout should be displaying "1". Touching both ends of the probes together should display "0.01" or a number very close to it. Next, connect one of the probes to the suspected ground port header and connect the other one to any shield casing on the device. Again, the multimeter readout should display "0.01" or a number very close to it if the port header is the ground.

For the remainder of this section, testing will need to be conducted with the device powered on. The next test is for the Vcc pin. Although it will have no further uses after this test, it is important to ascertain the embedded device operating voltage. With the multimeter set to 20 volts - direct current, connect one probe to the ground port header and the other to the suspected Vcc port header. The readout should be a steady 3.3 or 5 volts  $\pm 10\%$ .

Next, the TX port header can be tested. For this test, it is best to power off the device and power it back on as the serial port is more likely to be transmitting data via the serial connection during the boot up process. For the purpose of binary serial communication, 1's are represented as 3.3/5 volts and 0's as 0 volts (Jimb0, 2010). Therefore, the voltage in the TX header should be fluctuating rapidly between 3.3v and 0 volts. Given the speed of transmission, which is at least 300 bits per second and more likely several thousand bits per seconds, the multimeter will quickly resolve to displaying the average:  $\sim 1.6/2.5$  volts.

The last remaining port header will be the RX port header. Unlike the Vcc and TX port headers, there are no specific voltage levels to anticipate. While it would be logical to assume the voltage on the RX header to be 0 volts, each device tested during the research phase of this paper produced different voltage readings on the RX port header. Therefore, it is not possible to confirm if a port header is the RX port by simply using a multimeter. In the best circumstances, the port headers will be clearly marked. Otherwise, the incident handler will have to rely on the process of elimination to make an educated guess on which header is the RX port. In either case, it is highly unlikely that any damage will result from connecting the wrong port header to the incident handler's TX port as long as the voltages on both devices match within 10%.

### 2.3.3. Making the physical connection

When connecting two serial devices together, the transmit (TX) port on the first device must be connected to the receive (RX) port of the second, and vice-versa. This type of connection is known as a null modem (Hallinan, 2010). If both the embedded system and

the incident handler's computer are equipped with external serial connectors, then a null modem cables as shown in figure 3 on page 10 can be used. However, if either or both devices only have port headers located on the PCB, then jumper cables will be required.

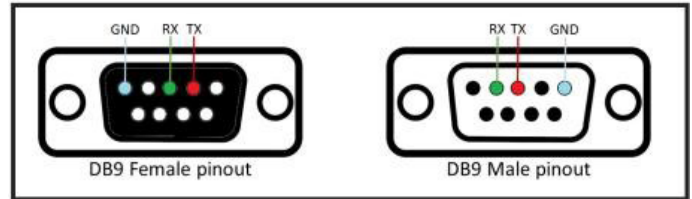


Figure 5: DB 9 Serial Connector Pinout. - (Source: Author)

If the embedded device is equipped with header pins, then a set of three male- female jumper wires can be used to connect to the USB-to-Serial adapter as shown in figure 6. If the device is equipped with header holes, then solderless pins can be used to create a temporary set of pins and the connection can be configured as per the previous example. In cases where the embedded device has header pads, then some soldering work may be required. Soldering is not that difficult and instruction videos abound on YouTube.com. But first, it would be a good idea to check out the blog titled "A solder- free UART connection<sup>6</sup>", which offers an imaginative alternative using spare pin headers

In cases where the embedded device has header pads, then some soldering work may be required. Soldering is not that difficult and instruction videos abound on YouTube.com. But first, it would be a good idea to check out the blog titled "A solder- free UART connection<sup>6</sup>", which offers an imaginative alternative using spare pin headers and a cloth pin. Regardless of the approach used to attach the jumper wires to the header pads, it is highly recommended to conduct a continuity test between each adjacent jumper wires to ensure that a short was not unintentionally created while securing the jumper wires to the PCB. Also, repeating the validation tests at the other extremity of the jumper wires for the GND and TX ports as described in section 2.3.2 will eliminate the probability of communication failure due to an imperfect connection between the jumper wire and the header pad.



Figure 6: Physical Connections to Serial Port Headers - (Source: Author)



## 2.4. Configuring the terminal console and connecting to the embedded device

### 2.4.1. Configuring minicom

Now that the devices are physically connected, it is time to configure minicom to be able to read the data being transmitted by the embedded device. The very first step is to enumerate the serial ports available on the incident handler's computer using the command **dmesg | grep tty**. In the example below, we can see the computer has both a standard serial port (*ttyS0*) and a USB-to-Serial Adapter (*ttyUSB0*) available.

```
root@kali:~# dmesg | grep tty
[0.000000] console [tty0] enabled
[1.218697] 00:06: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is
a 16550A
[9.672478] usb 3-2: pl2303 converter now attached to ttyUSB0
```

**Figure 7: Kali Terminal Window: Command to List Available Serial Connectors - (Source: Author)**

Next, we are ready to execute minicom and configure it to use the desired serial port. To do this, enter the command “**minicom -s**” in the terminal window to start the application and follow these steps once the minicom application is running:

1. Scroll down to the “*Serial port setup*” sub-menu item and press **ENTER**;
2. When the “*Serial port setup*” menu is displayed, press **B** to edit the Serial device;
3. Replace the serial device name with the one desired (*/dev/ttyS0*, */dev/ttyUSB0*, etc...); and,
4. Turn off all *Flow Control* features using the letters **F** and **G**.

The next step is to configure the communication parameters. Ideally, the parameters will have been discovered while researching the device hardware specifications as suggested in section 2.1.2. Otherwise, try 115,200 bps with 8N1 to begin with. While still in the “*Serial port setup window*”, press **E** to access the communication parameter menu. Enter the desired parameters and exit the menus by pressing **ENTER** until you return to the “*configuration menu*”. Then, it is highly recommended to save the configuration using the menu item “*Save setup as dff*”. This will ensure that any changes made will persist across application restarts and system reboots. Finally, exit the configuration menu and return to the minicom console.

### 2.4.2. Establishing and troubleshooting the connection

The incident handler's system is now ready and listening to the selected serial port. Powering up the embedded device will initiate the boot process, and soon after, data should be pushed on the screen

in the minicom terminal console. If there is no data, it is likely due to one of the following issues listed in decreasing order of likelihood:

**Table 1. Serial connection troubleshooting scenarios.**

Problem	Troubleshooting Approach
Wrong serial device ( <i>ttyS0</i> , <i>ttyUSB0</i> ) selected in minicom	<ol style="list-style-type: none"> <li>1. Reconfirm serial port name and availability using <b>dmesg   grep tty</b> command.</li> <li>2. Verify minicom “<i>serial port setup</i>”.</li> </ol>
Incorrect communication parameters	<ol style="list-style-type: none"> <li>1. Research device on the Internet.</li> <li>2. Sequentially try different parameters as described in section 2.4.3.</li> </ol>
Faulty wire or connection between the devices (i.e. cold solder)	<ol style="list-style-type: none"> <li>1. Verify all wires are securely fastened.</li> <li>2. Conduct connectivity test on all cables and wires.</li> <li>3. Revalidate the TX and GND wires as described in section 2.3.2 at the far end of the cable connected to the embedded device.</li> </ol>
Port headers on embedded device not a serial connection	<ol style="list-style-type: none"> <li>1. Redo the steps used to locate serial port headers.</li> <li>2. Research device on Internet for information on locating device's serial port headers.</li> </ol>
Serial port on embedded device disabled at the hardware level (practically unheard of as of this writing)	<ol style="list-style-type: none"> <li>1. Accessing the device through this approach will not work. Investigate other methods of accessing the OS and data such as telnet or ssh.</li> </ol>

If there is data displayed but it is garbled, then the embedded device is transmitting data that is being successfully received by the incident handler's computer.

### 2.4.3. Communication parameters discovery

If incorrect communication parameters is the suspected cause as to why minicom is not receiving the embedded device data output, then the only definitive way to find the correct parameters is to try each of them one after the other until meaningful data appears or all possible permutations have been exhausted. Although this will be a time consuming task, it is possible to increase the odds of finding the correct communication parameters quickly by following a few simple guidelines.

First of all, always start by using “8N1” for the data length, parity bits and stop bits parameters. Only change one of these at the time and only after having exhausted all possible speeds first. Second, begin with 115200 bps and then try 9600 bps and 38400 bps as these are by far the most common speed parameters. If unsuccessful, try all other possible speed starting with the fastest: 57600, 19200, 14400, 7200, 4800, 2400, 1800, 600, and finally 300 bps.

Also, it is best to power off the embedded device between each attempt as it is much more likely to transmit large bursts of data

while it goes through its boot-up sequence. Finally, it is best to exit minicom after each attempt and start is back up with the command: **minicom -s**. This will bring up the configuration menu and allow immediate access the “*Serial port setup*” sub-menu to alter the communication parameters following the steps described earlier in section 2.4.1.

## 2.5. Terminal console interaction

Eventually, the embedded device will boot and the forensic analyst’s computer will begin receiving data that it will display on the minicom terminal console. This is where a creative and resourceful incident handler will shine. Although there will be similarities between embedded systems, not all systems will immediately provide a terminal console with root access. Some research and analysis of the embedded device data output, along with a fair amount of trial and error, will likely be required to coax the device into relinquishing its secrets.

At this point, it is best to power off the embedded device, record the communication parameters and exit minicom. During the boot up process, a lot of information will quickly scroll through the terminal console; too much data in fact for anyone to comprehend in a single passing glance. To ensure that no invaluable piece of information goes unnoticed, it is best to restart minicom with the command line switch “- C” followed by a file name. This will ensure that all input and output are immediately saved to file for later analysis.

For the purpose of this paper, a refurbished Samsung Blu-ray Player, model BD- F5700, was procured online for \$40 USD and used as the test system. All examples are drawn from the capture of this Blu-ray player boot-up output and responses to commands entered from the incident handler’s computer.

### 2.5.1. Initial data analysis

Once minicom is running and is adequately configured to accept incoming data, it is time to turn on the embedded device. It will initialize and begin the boot process as explained earlier in section 2.1.3. After a quick pre-boot hardware initialization, the bootloader is loaded into memory and shortly after, the embedded Linux OS begins loading. This is evident by the data strings typical of most of Linux’s boot sequence screens. The data will continue flowing for up to a minute. But eventually, it will either stop or as is the case with the test system, the same five lines will repeat at regular intervals. This is a sure sign that the boot cycle has completed, and it is now time to dive into the capture file to tease out valuable nuggets of information. Below is a sample output of the capture file from the test system. The notation “...” represents one or more lines of data that have been removed to declutter the example.

```
preloader v.9773 CFG = 0x1
[0x00092000] [0x703fc000]
...
U-Boot 2009.08 (Jul 02 2014 - 10:57:04)

NXP B.V. - MT85XX SoC with ARM1176JZF-S
DRAM: 384 MB
...
Bootloader version 3847

Hit any key to stop autoboot: 0
## Booting kernel from Legacy Image at 0d9ffc0 ...
...
Starting kernel ...
...

Linux version 2.6.35 (yoseph@BD-Server-2) (gcc version 4.5.1 (GCC) )
#1 PREEMPT Wed Jul 2 10:56:36 WIT 2014
...
Kernel command line: root=/dev/ram0 rw initrd=0x16700000,0x001c50c5
console=ttyMT0 kgdboc=ttyMT0 mem=384M mt85xx_reserve=367M,17M
drvmem=227M,73MBL_Ver=3847
...
=== mt8551_init ===
...
<scrn_svr> app: home send msg =2...
{IOM} key value: 0xbf000
home_send_msg :i4_ret 0
...
<--- Start of repeating sequence,
Boot process is over
```

Figure 9: Test System Boot Process Output Summary - (Source: Author)

By looking at this output, it is immediately evident that there are key pieces of information displayed. First of all, the SoC Model number and ARM processor model number are clearly noticeable: “MT85XX SoC with ARM1176JZF-S”. A quick google search reveals that the SoC is manufactured by MediaTek<sup>7</sup>, and the ARM processor technical reference manual<sup>8</sup> is available online. These two sources of information can reveal much regarding the internal workings of the embedded system, including which distributions of Linux are compatible. Looking further down in the output, the exact model number of the SoC is revealed: “mt8551\_init”. This information is invaluable to understand the various subcomponents that may be accessible.

The test system also uses the “U-Boot 2009.08” bootloader. Furthermore, it appears that it is possible to stop the bootloader before the OS is loaded into memory and executed (“Hit any key to stop autoboot: 0”). This is obviously a very promising opportunity as it may allow the incident handler to interact with the system before the OS is booted. This particular line of investigation will be explored further in the next section.

Finally, the Linux version is displayed as the OS begins its boot process. The Kernel command line is also displayed, hinting that it may be possible to alter it from the bootloader menu. This possibility will be discussed in the next section. The OS Boot process and terminal console interaction with the OS will be discussed in further detail in section 2.5.3.

## 2.5.2. Bootloader analysis

It is now time to take a closer look at the bootloader output. As noted in the previous section, the test system bootloader is U-Boot 2009.08 version 3847. Das U-Boot, as it is officially named, is a very versatile bootloader distributed under an open source license and that has become very popular in the embedded Linux community (Hallinan, 2010). Das U-Boot has a very well documented online presence<sup>9</sup>, and a wise incident handler will quickly review the online documentation to get a sense of U-Boot's capabilities. Of particular note is U-Boot's ability to initialize and use the Ethernet port for IPv4 communications. U-Boot is also able to read and write to a USB drive.

Armed with some knowledge regarding the bootloader, it is time to reboot the embedded device. But this time, the incident handler should hit the "ENTER" key repetitively in an attempt to halt the bootloader before it starts loading the OS. This may take a few tries to succeed. The resulting output will look something like this:

```
preloader v.9773
...
U-Boot 2009.08 (Jul 02 2014 - 10:57:04)
NXP B.V. - MT85XX SoC with ARM1176JZF-S DRAM: 384 MB
NAND: ARM2 00:00:02.057 [FAST_LOGO] read flash
...
u-boot adaptive mtd mechanism applied. [_i_find_part_tbl]Part tbl info
passed from preloader [_i_find_part_tbl] version is 1!!
[NAND][Read_NoSkipBad]u4DevId = 0, u8Offset = 0x680000, u4MemPtr =
0x1efdf53, u4MemLen = 0x1 Bad block table found at page 131008, version
0x01 [NAND][read_awn_flag]uart_flag=0xff
...
Using default environment
In: serial
Out: serial
Err: serial
args_to_uboot:
head sig : 0xa0b0ead1
version : 1
boot type : 0x10000000
dram ch1 : 0x08000000
dram ch2 : 0x0d9ffffc0
kern addr : 0x16700000
initrd addr : 0x001c50d0
initrd size

enable bim two way write.
boot type:[0]
Bootloader version 3847
...
Hit any key to stop autoboot: 0
<<MTK:8551>>
mt8551_base # : 0x001c50d0
```

**Figure 10: Interrupted U-Boot Sequence Leading to Bootloader Console Prompt - (Source: Author)**

The prompt "mt8551\_base #" is the indicator that U-boot autoboot has been interrupted and it is now possible to interact through the terminal console with the bootloader. There are several important pieces of information displayed, including the memory address of the embedded Linux OS and the fact that both normal and error output is being sent to the serial console. Entering the command "help" will display a list of commands available to the incident handler.

```
mt8551_base # help
bdinfo - print Board Info structure
bootm - boot application image from memory
bootp - boot image via network using BOOTP/TFTP protocol
chpart - change active partition
fatinfo - print information about filesystem
fatload - load binary file from a dos filesystem
fatls - list files in a directory (default /)
mt85xx_boot- mt85xx_boot - boot command for mt85xx platform
nboot - boot from NAND device
ping - send ICMP ECHO_REQUEST to network host
printenv- print environment variables
rarpboot- boot image via network using RARP/TFTP protocol
setenv - set environment variables
tftpbboot- boot image via network using TFTP protocol
usbboot - usbboot - boot from USB device
```

**Figure 11: Sample output from "help" command - (Source: Author)**

The actual output contains far more commands. However, this paper will quickly highlight only a few of the most important ones. First of all, U-Boot provides several mechanisms to load and execute a Linux OS image from a number of sources including the NAND non-volatile memory, a USB thumb drive, and even across the network. This particular capability could be invaluable if a forensic investigator would wish to access the NAND memory using a custom built embedded Linux OS and then subsequently load the resident OS as a read-only partition for analysis. Other alternatives can include uploading the embedded Linux OS to another location for analysis. This can even be used to replace the existing embedded OS with a different or modified OS.

The bootloader also provides commands to display more valuable information about the embedded Linux OS. For example, the command "bdinfo" will display the boot parameters to be used as well as the device assigned IP address. Whereas the commands "printenv" and "setenv" can be used to view and modify the embedded OS boot arguments.

```
mt8551_base # printenv
bootcmd=mt85xx_boot_nand
autostart=yes
verify=no
bootdelay=0
baudrate=115200
ethaddr=00:0C:E7:00:00:00
ipaddr=192.168.0.124
gatewayip=192.168.0.1
netmask=255.255.255.0
loadaddr=0x2000000
...
stdin=serial
stdout=serial
stderr=serial

bootargs=root=/dev/ram0 rw initrd=0x16700000,0x001c50c5
console=ttyMT0 kgdboc=ttyMT0
mem=384M mt85xx_reserve=367M,17M drvmem=227M,73M BL_Ver=3847

mt8551_base # bdinfo
arch_number = 0x000007D0 env_t
= 0x00000000 boot_params =
0x00000100
DRAM bank = 0x00000000
-> start = 0x00000000
-> size = 0x18000000 ethaddr
= 00:0C:E7:00:00:00
ip_addr = 192.168.0.124
baudrate = 115200 bps
```

**Figure 12: U-Boot "printenv" and "bdinfo" outputs - (Source: Author)**

In this particular case, the boot command "mt85xx\_boot\_nand" is easily identified. As a result, it is possible to modify the Linux boot arguments to alter the behavior of the Embedded Linux at boot and then initiate the boot sequence. U-Boot's capability to easily define, store, and use environment variables makes it a very powerful tool in this area (Schocher, 2011).

This is but a quick overview of what can be done to the test system through the U-Boot console. An entire Gold paper could be written on the ways with which an incident handler could interact with an embedded system using only a bootloader such as U-Boot.

### 2.5.3. Embedded Linux OS Analysis

Continuing on with the boot sequence, the OS begins to send line after line to the minicom terminal console. This is where a good comprehension of the Linux environment will help the incident handler pick out invaluable data.

```
## Booting kernel from Legacy Image at 0d9fffc0 ...
Image Name:
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:   1528968 Bytes = 1.5 MB
Load Address: 0da00000
Entry Point: 0da00000
Loading Kernel Image ... OK

OK

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
Linux version 2.6.35 (yoseph@BD-Server-2) (gcc version 4.5.1 (GCC) )
#1 PREEMPT Wed Jul 2 10:56:36 WIT 2014
...
[kernel zone size]DMA: 61440KB, NORMAL: 304128KB, MOVABLE: 27648KB
...
Kernel command line: root=/dev/ram0 rw initrd=0x16700000,0x001c50c5
console=ttyMT0 kgdboc=ttyMT0 mem=384M mt85xx_reserve=367M,17M
drvmem=227M,73M BL_Ver=3847
...
Memory: 384MB = 384MB total
Memory: 291896k/291896k available, 101320k reserved, 0K highmem
...
NAND device: Manufacturer ID: 0x2c, Chip ID: 0xda (Micron NAND 256MiB
3,3V 8-bit)
Creating 23 MTD partitions on "NAND 256MiB 3,3V 8-bit":
0x000000000000-0x000000200000 : "boot_1"
0x000000200000-0x000000400000 : "part_info_1"
0x000000400000-0x000000600000 : "part_info_2"
...
x000001400000-0x000001600000 : "initrd_1"
0x000001600000-0x00000aa00000 : "rootfs_normal_1"
...
0x00000c400000-0x00000f620000 : "ubi0"
...
hard sector size is 512
devblksize is 4096
...
INIT: version 2.88 booting
star: rx descriptor idx:10 forINIT: Entering runlevel: 5
=rc5 Start=
...
```

Figure 13: Embedded Linux boot sequence summary - (Source: Author)

First of all, the output displays the kernel command line arguments used. It is particularly useful if the U-Boot “setenv” command was used as described in the previous section to verify that the modified argument(s) were successfully passed to the OS. Afterward, the OS begins initializing the non-volatile memory and creates “MTD” partitions. The Memory Technology Device (MTD) subsystem for Linux provides access to non-volatile memory storage, typically Flash devices (Woodhouse, n.d.). MTD provides the mechanisms for putting fully functional file systems into Flash, which can be read from as well as written to. In addition to indicating the various partitions loaded, this data output also provides mapping to the flash memory address, in effect providing the information necessary

to mount these partitions from an OS of the incident handler’s choice as suggested in the previous section.

Finally, the output reveals the OS is entering “runlevel 5”. This is a standard runlevel of a Linux system and by the looks of the follow-on output, it starts the initialization of the test system’s multimedia functionality as well as the on-screen display user interface. This generates more output to the terminal console, which also yields interesting information.

```
=rc5 Start=
...
home network app update 1 US
Opera TV Store is enable
Netflix is enable
...
flickr is enable
Facebook is enable
...
>>>>>>>>>>a network init >>>>>>>>
COMM_FUNC_NETWORK>>>a_network_wlan_task_reg_cbk >>> [WIFI_MW] [WPA]
Default
[Enter]c_net_wlan_wpa_reg_cbk :

ap_scan=1
<get wifi> Bssid is : ff:ff:ff:ff:ff:ff
SSID:HOME-A4A7-2.4, SsidLen:13, eAssocCase:2, eAuthMode:17, e_
AuthCipher:24 Priority:0
wlan_favorite AP is found
KeyIsAscii:1 KEY: ThisismySecretPassword!
[WIFI_MW] [WPA] Default [Enter]c_net_wlan_associate :
...
> GET /openapi/conf/version HTTP/1.1 Host: www.samsungotn.net
Accept: */*
AppKey: bdf9c5fc-cd9d-11d3-95b9-10000040004-08b30ecb-f578-4ebe-b020-
07dc6acaf82f
IPAddr: 10.0.0.176
Token: 85dd4d2b+ F5700
WW_664c7452149e+5e316cb3fde6b74e909ded320bb6b2791496d821
...
< HTTP/1.1 200 OK
< Cache-Control: private
< Content-Type: text/xml; charset=utf-8
< Date: Wed, 01 Apr 2015 00:59:36 GMT
< Connection: close
<
<UPG> upg_get_configuration_xml_file_cb

=====http notify status : 0
=====
<UPG> upg_parse_configuration_xml_file
<UPG> rsp ok
upg_str_replace() before replace---str_src:P8HNDV:${SecKey}
upg_str_replace() after replace---str_src:P8HNDV:ccc935ce-81b7-40b3-
94e3-8f1bc65e315b-4659c7ab-3861-4055-a7dc-ed6b4fa5d0cf
<UPG> g_str_murl: http://www.samsungotn.net/openapi/tv/F5700_WW/BSP-
F5700WWB-/m_notice
<UPG> g_str_passwd: P8HNDV:ccc935ce-81b7-40b3-94e3-8f1bc65e315b-
4659c7ab-3861-4055-a7dc-ed6b4fa5d0cf
<UPG> upg_get_pdl_xml_file
> GET /openapi/tv/F5700_WW/BSP-F5700WWB-/m_notice HTTP/1.1 Host: www.
samsungotn.net
Accept: */*
DUID: BDCL6GDVURUQMM
```

Figure 14: Sample Embedded Linux runlevel 5 console output - (Source: Author)

Of particular interest is the fact that there are 13 applications installed and running. This includes the likes of Netflix, Pandora and Twitter. Any one of which could be a likely vector of infection if the incident handler is investigating the suspected presence of malware. Even more interesting is the fact that the output lists all previously associated wireless access point including the password keys in the clear! Finally, the output begins displaying HTTP

traffic with Samsung servers for the purpose of verifying if the firmware is up to date. This appears to require the use of password strings that are again, displayed in the clear. This is interesting because an attacker, if successful with DNS redirection, may be able to replace the embedded device firmware with code of his choosing. This is definitely something the incident handler will want to investigate. Or use himself as an alternate method to inject a firmware of his own.

At some point, the outpouring of data to the terminal console will stop. In many instances, this is where the incident handler finds himself face to face with a Linux shell with root privileges. If this is the case, then he can immediately begin gathering information in the same way he would with any Linux PC. However, as is the case with the current test system, the developers may have programmed in a custom shell to facilitate their needs but also constrains access to some degree. In such cases, the incident handler will have to experiment and see what can be done to elevate his access privileges. Or at least, he would have to collect the information he needs through the custom shell. With the test system, it was possible to invoke a password prompt by sending a **CTRL-C** command to the system. Simply pressing **ENTER** returned what appeared to be a list of available commands with a short description.

```
Command> [CTRL-C]
01/01/2010 00:16:00 *
Password: Access denied!
Oops! you are having a trouble, try again...

Command> [ENTER]

[Help]
basic(b):      Basic
              Middleware
mw:           Multimedia Middleware
mmw:          MTK tool
mtktool(0):   Set uart baudrate
setbaudrate(setbr):

Command>mw
[Help]
mkfs:         FAT16 Format
mount:        Mount a filesystem
umount:       Unmount a filesystem
dir:          List files on specified directory
...
```

Figure 15: Embedded Linux custom shell sample - (Source: Author)

It may be possible to get past the password prompt though simple password guessing. Also, a dictionary attack against the password prompt using minicom's "runscript" script interpreter could be attempted. On the other hand, the commands available through the custom shell may be able to yield most if not all of the information sought by the incident handler. The best approach to investigate such a custom shell is to build a mind map as shown in Appendix A. Then, the incident handler could develop a meticulous plan to investigate the most promising commands.

A thorough analysis of the test system custom shell is beyond the scope of this paper. However, two lines of investigations will be briefly discussed to demonstrate the potential of this approach.

First, a close look at the "ave\_tcp" sub-command reveals numerous networking tools. This includes tools to display the device IP address (`dhcpc_get_info ip_info`). It also includes tools to resolve IPs and ping hosts on the network (`hostname hn`), (`dns_lookup dns_lk`), (`ping p`), (`pinghostbyname p_host`). And most interestingly, it also provides the mean of enabling a telnet daemon on the embedded device using the command `invoke_telnetd td`. Having executed this command, the test system indeed had a telnet daemon listening and it was possible to connect to it from the incident handler PC using PuTTY. Unfortunately, the console demanded a login name and password before allowing any further access.

The custom shell also provided a very elaborate set of file management features including the ability to format a partition, mount and unmount partitions, copy and compare files, and much more. But perhaps the most interesting features are the abilities to list directory contents and read files. These two commands alone enable the incident handler to conduct a complete reconnaissance of the file system looking for data of interest. Figure 16 and figure 17 provides directory listings of some of the more interesting folders.

<pre>Command&gt;mw.fm.dir . .. null) var usr tmp sys sbin res proc plugins mnt misc lib init etc cust_part_1 cpm bin acfg root dev</pre>	<pre>Command&gt;mw.fm.dir /etc . .. hosts resolv.conf init.d Wireless wifi.script nsswitch.conf fstab inittab protocols host.conf passwd hostname group mtab services DfbkeyMapToQtkey inetd.conf</pre>
--	---

Figure 16: Directory listings using the command "dir" - (Source: Author)

<pre>Command&gt;mw.fm.dir /etc/init.d . .. upg_micro_be.sh upg_2.sh upg_1.sh rc.fast_shutdown upg_prog mtd_init.sh upg_prog.sh usb_init.sh rc0 rc.shutdown mtd_ubil_init.sh rc.standby rc5 rc.fast_reboot rcSinit rc6 rc5 rc.reboot</pre>	<pre>Command&gt;mw.fm.dir /mnt . .. nand_06_0 nand_03_0 log ubi_boot rootfs_enc rootfs_normal rootfs_it rootfs_enc_it  Command&gt;mw.fm.dir /tmp . .. P2P_DEV_CONF WPA_CONF browser mtkcfg mtkpbmisc mtkpbsnd mtkpbctrl</pre>
---	---

Figure 17: More directory listings using the command "dir" - (Source: Author)

It is possible to gather more information on the system and the OS using the “read” command to access the various data elements in the “/proc” directory such as *cpuinfo*, *version* and *mounts*. However, the output is in hexadecimal. Therefore, some conversion will be required to read the output. Thankfully, this can be easily remedied using RapidTables.com Hex to ASCII converter<sup>10</sup>. It is also possible to read and even modify *runlevel* scripts. Although there is no built-in text editor, it is still possible to edit or even replace the script using the “cp” command to overwrite the file. Finally, Figure 17 shows that the file “/etc/passwd” is listed but unfortunately, the file “/etc/shadow” is missing. This could be a sign that the custom console is running with restricted privileges. Nevertheless, it is worthwhile to look at the “/etc/passwd” file to see what user accounts are available. Figure 18 below demonstrates the command to display the “/etc/passwd” file and the resulting output conversion into ASCII.

```
Command>mw.fm.read /etc/passwd 0 0 1000 0
read file</etc/passwd>, from: 0, offset: 0, cnt: 1000, w/wo cache: 0
/etc/passwd size is 66 bytes
/etc/passwd blk size is 4096
/etc/passwd blk cnt is 8
ui4_read_cnt: 66
00000000h : 72 6F 6F 74 3A 24 31 24 42 4A 4C 51 39 6E 34 4E
00000010h : 24 32 63 43 6C 47 2E 7A 54 78 78 53 5A 33 54 6D
00000020h : 4A 72 45 70 48 4C 2E 3A 30 3A 30 3A 72 6F 6F 74
00000030h : 2C 2C 2C 3A 2F 72 6F 6F 74 3A 2F 62 69 6E 2F 73
00000040h : 68 0A 00 00 00 00 00 00 00 00 00 00 00 00 00
Hex to ASCII converter
Enter 2 digits hex numbers with any prefix / postfix / delimiter and press the Convert button:
72 6F 6F 74 3A 24 31 24 42 4A 4C 51 39 6E 34 4E
24 32 63 43 6C 47 2E 7A 54 78 78 53 5A 33 54 6D
4A 72 45 70 48 4C 2E 3A 30 3A 30 3A 72 6F 6F 74
2C 2C 2C 3A 2F 72 6F 6F 74 3A 2F 62 69 6E 2F 73
68 0A
Convert Reset
root:$1$BjLQ9n4N$2cClG.zTxxSZ3TmJzEpHL.:0:0:root,,,:/root:/bin/sh
Select
```

**Figure 18: /etc/passwd read from custom shell and converted into ASCII**  
- (Source: Author)

The result is a pleasant surprise. The file not only contains a single username. Which suggests the custom shell is indeed running with root privileges. But it also contains the hashed password. Yet another avenue of analysis where the incident handler may choose to run a password cracking tool such as hashcat11 to conduct a dictionary attack or bruteforce the password and gain root console access via the “CTRL-C” input directly at the serial console or thought the telnet console discovered a few paragraphs earlier.

### 3. Conclusion

With the rapidly growing number of embedded devices found virtually everywhere, and recent indications of such devices having been compromised and some even used in botnets, it is only a

matter of time before one of them becomes a key link in your incident investigation. However, an embedded device does not need to be thought of as an insurmountable obstacle from which evidence collection is impossible. With a basic appreciation of embedded systems architecture, a decent understanding of the boot process, and a good grasp of Linux, it is possible to access key files located on embedded systems that can potentially hold invaluable evidence for the investigation.

The serial port is one of the oldest technologies available in embedded systems today. Because of its simplicity and ease of use, it is the interface of choice for system developers, allowing them to easily read messages from and interact with the system during boot and normal operations. As a result, they can also be used by an incident handler to access the inner workings of an embedded system. But, serial ports have gone through some improvements over the years to better suit the needs of embedded systems. Therefore, not only is it important to correctly identify the serial port headers on a PCB, but the incident handler must also be able to determine the voltage used to avoid damaging the embedded device or the PC used to connect to it.

Once physically connected, it becomes possible to interact with the system bootloader to image, copy, replace, or alter the embedded Linux OS and/or non-volatile file storage. At the very least, access to the bootloader permits the alteration of the boot parameters in a manner favorable to the incident handler. Also, many embedded devices serial console will immediately present a root shell. Others will offer a shell with limited capabilities. However, as demonstrated in this paper, even with a limited shell it is still possible to access the most critical areas of the file systems up to and including the “passwd” file.

In conclusion, this research project demonstrates an overview of the potential actions an incident handler may take when investigating an embedded system OS. In practice, the extent of the investigation is really limited by the breath of the handler’s experience and his imagination. All it really takes to succeed is a determined incident handler with a sound understanding of the technologies involved, patience, an ability to think critically, and a structured approach for the door to the inner workings of embedded systems to be opened.

### REFERENCES

- [1] Barry,P., & Crowley, P. (2012). Modern'Embedded' Computing:'De-signing'Connected,' Pervasive,'Media9Rich'Systems. Waltham, Mass: Morgan Kaufmann.
- [2] British Broadcasting Corporation BBC. (2014), January 17 . Fridge'sends'spam' emails'as'attack'hits'smart'gadgets. Retrieved from bbc.com: <http://www.bbc.com/news/technologyL25780908>
- [3] Electronic Industries Alliance EIA Standards. (1969) . RS232'Spec-ifications'and' standard. Retrieved from lammertbies.nl: [http://www.lammertbies.nl/comm/info/RSL232\\_specs.html](http://www.lammertbies.nl/comm/info/RSL232_specs.html)
- [4] Ganssle J., Noergaard, T., Eady, F., Edwards, L., katz, D., Gentile, R., . . . Huddleston, C. (2008), March 26 . Embedded'Hard-ware'Know'it'all. Elsevier: Newnes. Retrieved from BeyondLogic.com

- [5] Gartner. (2014), November 11 . Gartner Says 4.9 Billion Connected "Things" Will Be in Use in 2015. Gartner'Newsroom. Retrieved March 17, 2015, from <http://www.gartner.com/newsroom/id/2905717>
- [6] Hallinan, C. (2010) . Embedded'Linux'Primer:'A'Practical'Real-9World'Approach. Crawfordsville, Indiana: Prentice Hall.
- [7] Jimb0. (2010), November 23 . RS9232'vs.'TTL'Serial'Communication. Retrieved March 19, 2015, from sparkfun.com: <https://www.sparkfun.com/tutorials/215>
- [8] Lackorzynski, A., & Godisch, M. n.d. . Project'Info. Retrieved March 18, 2015, from Minicom Project Page: <https://alioth.debian.org/projects/minicom/>
- [9] Rothman , M., & Zimmer, V. (2013), May 29 . Using'UEFI'in'embedded'and'mobile' devices. Retrieved March 24, 2015, from LinuxGizmos.com: <http://linuxgizmos.com/usingLuefiLinLembeddedLandLmobileLdevices/>
- [10] Schocher , H. (2011), October 19 . 7.4.'Boot'Arguments'Unleashed. Retrieved March 24 2015 from DENX ULBoot and Linux Guide DULG : <http://www.denx.de/wiki/view/DULG/LinuxBoot-Args>
- [11] Waqas, A. (2010), October 28 . What'Is'Bootloader'And'How'To'Unlock'Bootloader'On' Android'Phones'[Complete'Guide]. Retrieved 03 27, 2015, from addictivetips.com: <http://www.addictivetips.com/mobile/whatLisLnlockLbootloader-LonLandroidLphonesLcompleteLguide/>
- [12] Woodhouse, D. n.d. . General'MTD'documentation. Retrieved March 31, 2015, from Memory Technology Device HTD Developers page: <http://www.linuxL.mtd.infradead.org/doc/general.html>

## ABOUT THE AUTHOR

**Eric Jodoin** is a cyber operations subject matter expert working for Canada's Department of National Defence (DND). He has accumulated 27 years of experience planning and conducting military operations in the Maritime, Aerospace, and Cyber domains. He began specializing in cyber security in 2002 and obtained his CISSP in 2009. He was once employed as a Cyber Domain Chief for the NORAD-USNORTHCOM Command Center and completed a tour as the Director of Operations for the Canadian Forces Network Operations Center. Eric holds a Master of Science Degree in Information Security Engineering (MSISE) from the SANS Technology Institute and holds several GIAC Certifications including GSE, GSEC, GCIA, GCIH, GCFE, GCFA, GWAPT and GPEN.



## Call for Papers for Publication

CSIAC is chartered to **leverage the best practices and expertise from government, industry and academia** in order to **promote technology domain awareness and solve the most critically challenging scientific and technical (S&T) problems.**

As part of that mission, CSIAC publishes the *Journal of Cyber Security and Information Systems*, focusing on scientific and technical research & development, methods and processes, policies and standards, security, reliability, quality, and lessons learned case histories. Contributing authors realize the benefits of being published in a highly-respected resource within the technical community, with an enormous reach across the Department of Defense and the broader scientific community (23k+ subscribers). Demonstrate your expertise and accomplishments or pose the challenging questions for further thought in the resource that reaches fellow Subject Matter Experts (SMEs) developing the solutions to support the warfighter.



Articles in the areas of **Information Assurance, Software Engineering, Knowledge Management, Information Sharing, and Modeling & Simulation** may be submitted.

## To Submit an Article

Visit:  
<https://www.csiac.org/csiac-journal-article-submission-policy/>


# OFFENSIVE INTRUSION ANALYSIS:

## Uncovering Insiders with Threat Hunting and Active Defense

By: Matthew Hosburgh, Graduate, SANS Technology Institute, MS in Information Security Engineering







**Today's adversaries are advanced and more capable than ever before. Passive defensive tactics are no longer viable for pursuing these attackers.** To compound the issue, the existence of an insider threat creates a challenging problem for the passive defender. One of the largest breaches of classified information was carried out by an insider. Months after the incident had occurred, the Department of Defense (DoD) only began to realize the implications of the leak. The damage did not solely rest with the United States. A cascade of consequences was felt in many parts of the world, resulting from this breach. Techniques like Threat Hunting attempts to diminish this problem by combating advanced threats with people, also known as Threat Hunters. Although Threat Hunting is proving to be invaluable for many organizations there remains a chasm between detection and disclosure. Offensive Countermeasure tools such as the Web Bug Server and Molehunt can be leveraged as a means to proactively hunt insider threats. To keep up with the continually evolving human adversary, defenders must employ these offensive tactics to annoy and attribute their adversaries.

## 1. Introduction

The words on a slide describing WikiLeaks 1.0 in late 2009 foreshadowed a grave and inevitable future. "...The leading disclosure portal for classified, restricted or legally threatened publications. We provide an anonymous safe harbour for the submission and uncensorable provisioning of documents" ("WikiLeaks Release," 2009). At the time, this statement was alarming to many individuals and organizations within the Department of Defense. In 2010, over 391,000 classified U.S. documents were leaked by WikiLeaks which was the largest unauthorized disclosure of classified information to date (Romero, 2010). After the initial shock and awe of the leak had subsided, specific documents and information surfaced out of the massive trove of documents. In Tunisia, the U.S. information pointed to greed and corruption of the Tunisian government, which helped fuel the Arab Spring (Bachrach, 2011). The effects did not stop there. The protests in Tunisia had a cascading effect felt around the world. In New York, protestors were galvanized by the actions in Northern Africa and eventually Occupy Wall Street was sparked (Saba, 2011). But why and who was to blame? Months after the information was posted to WikiLeaks, an Army private was indicted as a suspect and sole actor. At the time, his privileged access to the material enabled his actions to expose the wrong doing, and much more, by the U.S. military during the Iraq War. This disparity from the point of breach to the moment of detection is still problematic. Techniques like Threat Hunting, attempt to diminish this problem by combating advanced threats with people, also known as Threat Hunters. Although these techniques are proving invaluable to many organizations, there remains a delta between detection and compromise. Attribution is an Active Defense technique that, when combined with Threat Hunting, is a method to drastically reduce the detection delta and to minimize the effects of a targeted attack.

Tools such as the Web Bug Server and Molehunt can be leveraged as force multipliers when hunting insider threats.

## 2. The Detection Delta

Detecting threats and adversaries on networks continues to be a problem for many organizations. In the 2017 M-Trends report by FireEye, "the global median time from compromise to discovery has dropped significantly from 146 days in 2015 to 99 days in 2016" ("M-Trends," 2017). This disparity is known as the detection delta. Although positive, the number still indicates that it takes over three months before an organization realizes they have been breached. Significant damage and data exfiltration can happen in 99 days. Put another way, 99 days is equal to  $8.554e+6$  seconds. At dial-up speeds of 56Kbps, that means an attacker could transfer approximately 59.87GB of data, assuming a constant bandwidth and connection. If an average customer record is 2KB in size, the total records lost would equate to 29,935,000—even at low and

slow speeds. Adding bandwidth or multiple avenues for the attacker to exfiltrate the data only exacerbates the loss to the organization. These numbers are daunting and almost impossible to comprehend. Traditional alerting further adds to the exhausting task of reactive detection techniques.

### 2.1. Alert Fatigue

Alert fatigue is an enemy to detecting or hunting real, human adversaries on an organization's systems. In a recent study on Computer Security Incident Response Teams (CSIRT), researchers discovered that many operators or analysts are not well prepared in terms of tooling: "All are uniformly unhappy with current solutions for forensics and incident response. Commercial solutions like Security Information and Event Management (SIEM) systems do not address operational needs, probably because vendors and researchers do not understand how analysts think and work" (Sundaramurthy, McHugh, Rajagopalan, & Wesch, 2014). This discontentment erodes at the trust of the alerts that an analyst receives. The alerts produced by varying tools are not useless; however, they can be overwhelming and time consuming. The study went on to discover that repeatable tasks were not being automated. The perpetual cycle erodes at the analyst's mental well-being: "Receive an alert, scan the logs (three minutes), look up an address (one minute), find the user information (another minute), repeat" (Sundaramurthy et al., 2014). The argument can be made that all work and no mental stimulation can make the analyst a dull boy, or girl. This might not be such a problem if all of an organization's adversaries were robots. The reality is that there are human adversaries with human behaviors and human flaws attacking organizations.

### 2.2. The Human Adversary

At the other end of any bot, virus, or targeted attack there is a human. Someone to code an action, someone to conduct reconnaissance on a target, and often, someone to exfiltrate an organization's protected information. According to research from Carnegie Mellon University, "the human domain is complex, and as a result the reasons behind certain behaviors are inherently complex" (Costa et al., 2016). This problem that many detection systems try to solve is the automated detection of these complex actions. Some of the actions are obvious, like an NMAP port scan. Others are less overt, such as valid credentials used for nefarious purposes. To compound the issue, not all humans or analysts use the same techniques or methods to achieve their goals. For example, a nation state actor could have a set of known techniques tactics and procedures (TTPs) that could potentially be detected. What if those TTPs change mid-mission? Or even more frustrating, what if an insider was operating in the parameters of a company policy to exfiltrate data? The detection delta grows and might even be non-existent in the case of an insider leaking information until the damage is done.

### 2.3. Common Threads via an Intrusion Model

Leveraging known data on attack techniques is an excellent starting point for advanced adversary detection. One such example is the MITRE Adversarial Tactics, Techniques & Common Knowledge (ATT&CK) matrix. MITRE's claim is that "ATT&CK is a constantly growing common reference for post-compromise techniques that brings greater awareness of what actions may be seen during a network intrusion. It enables a comprehensive evaluation of computer network defense (CND) technologies, processes, and policies against a common enterprise threat model" ("Adversarial Tactics," n.d.). Common attack patterns provide a start; however, they are still too broad to begin a Threat Hunt. Fortune might favor a pattern search to uncover an attacker, but the advanced adversary's actions will more than likely remain undetected. Historical references are another key area to investigate what is known about insider attacks.

Research by the Carnegie Mellon University provides an additional resource for developing patterns for hunting Insiders. Costa et al., (2016) analyzed data from the MERIT insider threat database, which contains instances of insider incidents. The research illustrated the insider's actions mapped out in an ontology model. Similar to developing patterns, this method hones in on the actual human behaviors. Each of the scenarios could be used to develop additional patterns to match on. Figure 1 is an example provided by Costa et al., (2016) which models the unauthorized exporting of confidential data by an insider with a laptop.

Because each organization is unique, a look at who the adversaries are and what their goals are is necessary in prioritizing the work of a Threat Hunter.

### 2.4. Prioritization of Adversaries

Two of the most fundamental questions an organization can ask are: what are we protecting and who are our adversaries? These two questions help to shape the larger security strategy, but can especially hone the focus of a Hunt Team. Because not all organizations are created equally, the answers will vary from industry to industry and even organization to organization within a common commerce. One of the most rapid and effective means to capture who the adversaries are, is via threat modeling. In the most rudimentary example, a simple survey polling the current staff can illuminate a solid list of potential, or known adversaries. The tribal or tacit knowledge is powerful because it is the collective body of knowledge that has been learned over many years. Often it is the assumed knowledge, or information that did not make it into a formal document. One such example of this analytical model is the Crown Jewels Analysis (CJA) Process. CJA "can lead the hunter to think about the most useful types of data to collect in the environment (and the locations from which it should be collected) to be able to begin hunting for types of adversary activity that might be especially important to detect" (Lee & Bianco, 2016). Knowing what requires protection ensures the focus is on the most meaningful areas of the organization.

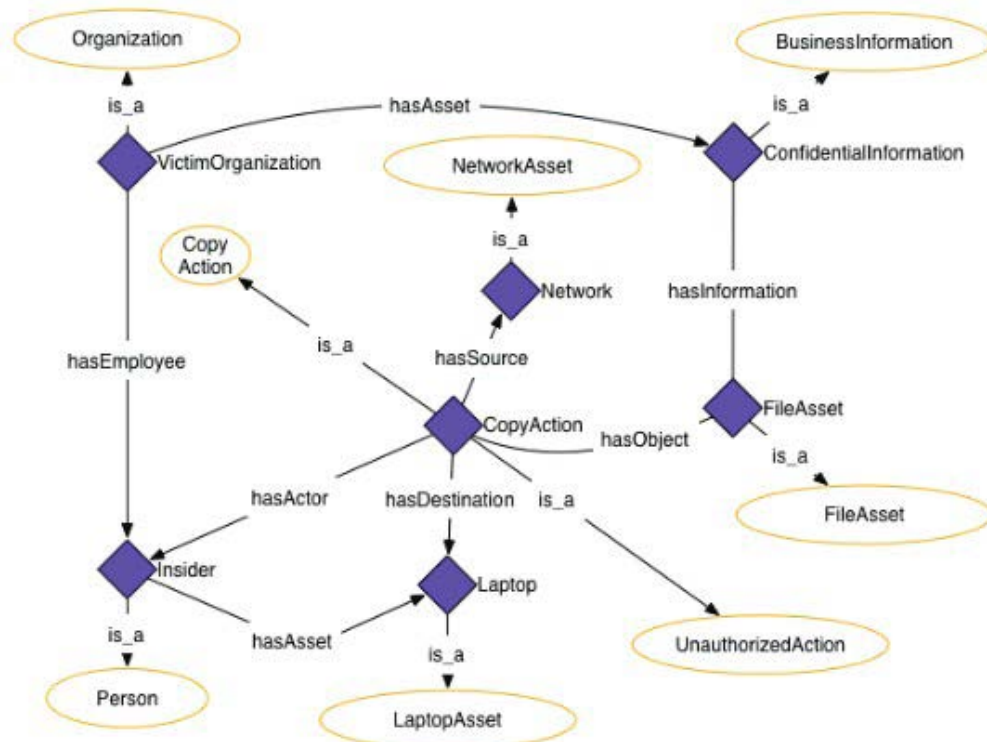
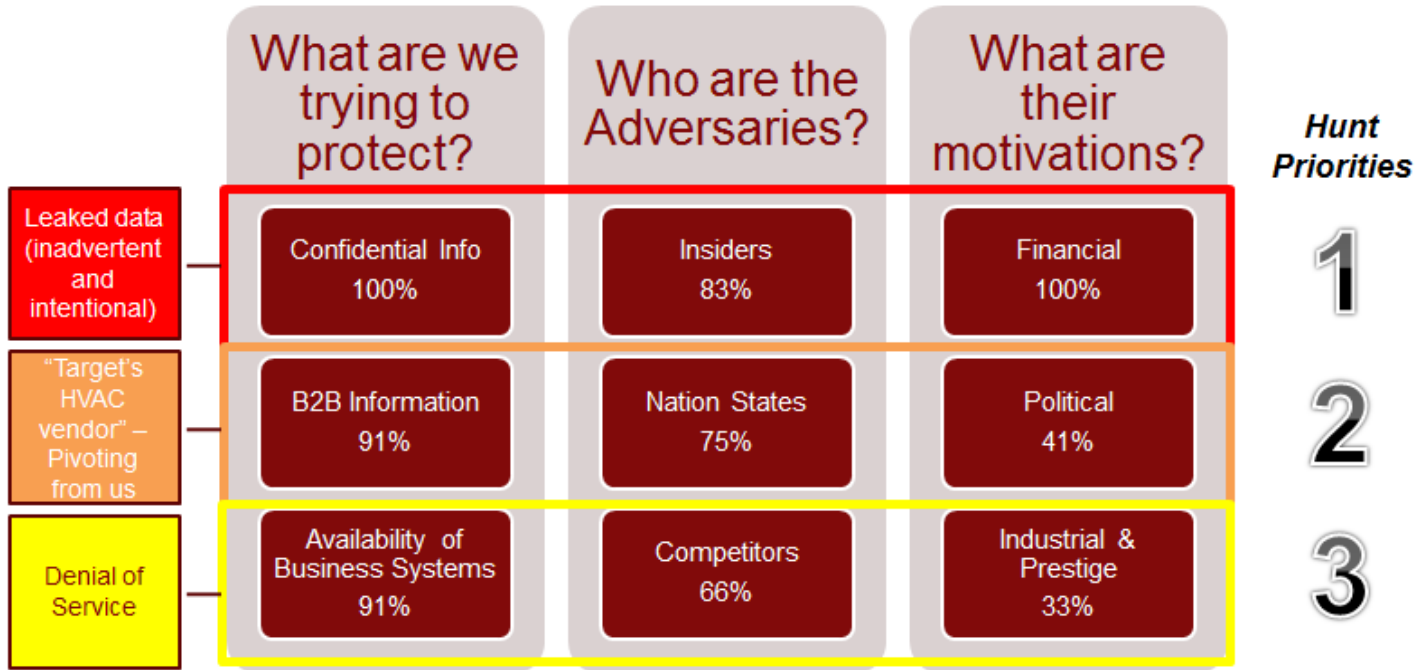


Figure 1: Ontology Model for Unauthorized Data Export - (Source: Author)



*These categorizations are not exclusive!*

Figure 2: Adversary Survey Results to Begin to Prioritize Hunting Actions - (Source: Author)

Looking at who the adversaries are can also be extracted from the tacit knowledge and reporting from the larger community. Based on these findings, the hunt priorities or intrusion analysis focus can be set forth. Geopolitical and other market factors help to further paint the adversary picture by helping to understand the actor’s motivation. Figure 2 illustrates a hypothetical model based on survey results.

The more of a cross-section within the organization, the comprehensive the results will be. Appendix A lists a series of questions that can be used as a basis for a survey.

### 3. Wait, What is Threat Hunting?

Threat Hunting can be defined as “the [proactive] pursuit of abnormal activity on servers and endpoints that may be signs of compromise, intrusion, or exfiltration of data [—both from external and internal entities]” (Gregory, 2017). To note, servers can include Windows, Linux, appliances, network devices, or modules that are acting to serve up a resource. An endpoint can be a laptop, mobile device, or other system that the proverbial user interacts with. True Threat Hunting is the area just beyond the automated detection capabilities of an organization. Simply put: it is the point where the human analyst or Threat Hunter must make the call on whether or not there has been a compromise, devoid of a definitive alert. Figure 3 illustrates the entire detection strategy that can be utilized. The more manual the detection area, the more skilled the Hunter must be.

Not all hunts can produce indicators of compromise, but when possible, it is the area where the human Hunter leverages automation to assist with both behavioral and atomic types of detection. For the biggest return, hunting and incident response need to work together.

#### 3.1. A Note on Incident Response

Incident response (IR) is a necessary component of Threat Hunting. According to Gartner, “Hunting success relies on a mature security operations center (SOC) and cyberincident response team (CIRT) functions” (Chuvakin, 2017). This is often true; however, it is not an absolute requirement to hunt. A mature organization might boast in having a robust set of procedures on how to handle malware, Denial of Service, and other attacks in place. A new organization, or a new response team, might have only a generic response plan. Regardless of the level of maturity, without some processes in place, hunting becomes a high fidelity alerting regime. The bigger value is achieved with hunting and IR working harmoniously. In Figure 4, the relationship between IR and Threat Hunting is shown.

When possible, Indicators of Compromise (IOCs) should be worked back into the automated detection system. Future alerts and detection patterns would trigger the IR process and not necessarily the Hunter. One such means to identify active adversaries is with the application of Active Defense, or Offensive Countermeasures.

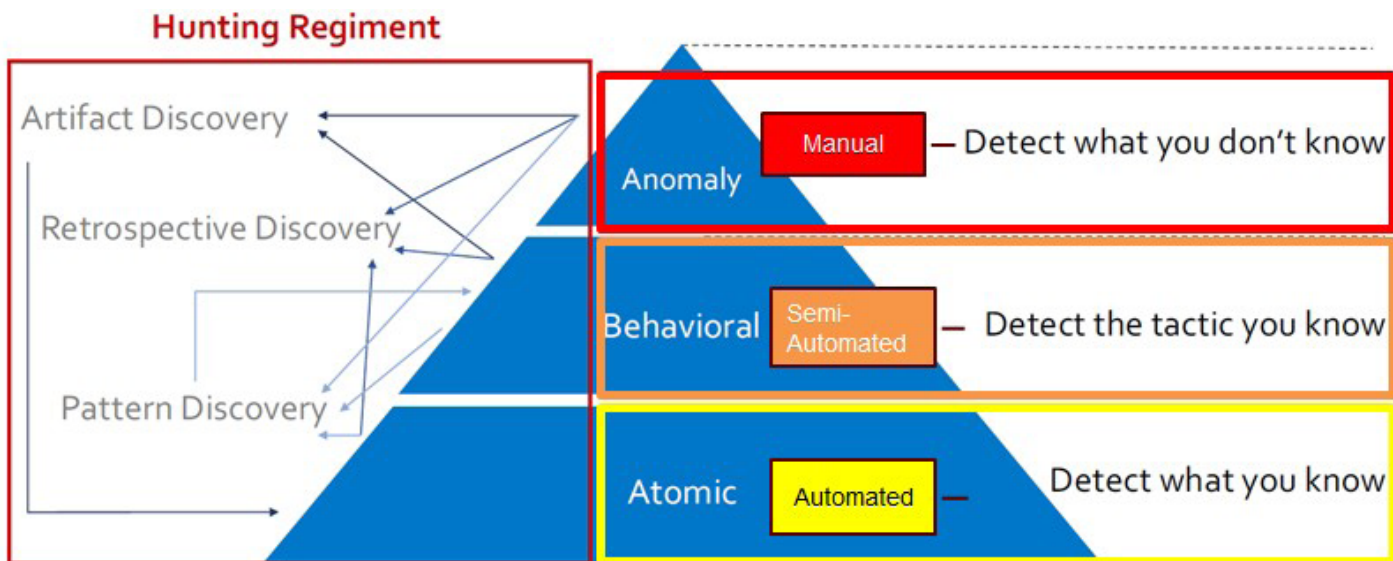


Figure 3: The Hunting Regiment in Relation to The Organization's Detection Strategy (Merritt & Concannon, 2017) - (Source: Author)

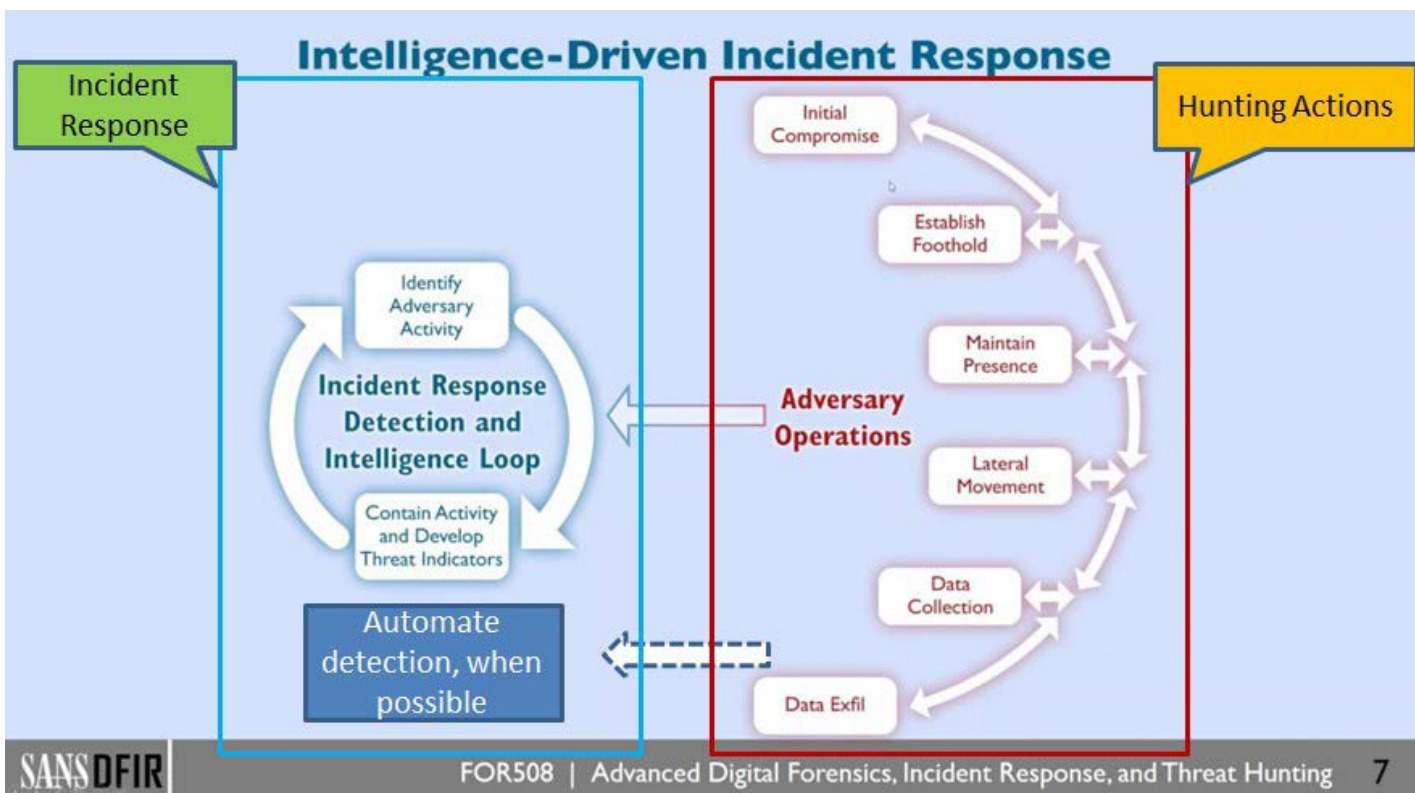


Figure 4: Threat Hunting and the Incident Response Relationship (Lee, 2016) - (Source: Author)

### 4. Offensive Countermeasures in the Hunt

In the pursuit of a human adversary, Offensive Countermeasures can act as a force multiplier in traditional Threat Hunting operations. Offensive Countermeasures are a set of techniques that can be leveraged to proactively pursue adversaries. The countermeasures focus on three Active Defense categories, referred to as the three A's (AAA). They are: Annoyance, Attribution, and Attack (Strand, Asadoorian, Robish, & Donnelly, 2013). Attribution will be the focus and primary method to hunt for active insider threats. Strand et al. (2013) provides its definition when they say, "Attribution is focused on knowing who is attacking you" (2013). As simple as it may sound, illuminating who is attacking an organization is a challenging endeavor. Challenges such as virtual private networks (VPNs), compromised hosts being used as an attack platform, proxies, and other obfuscation techniques help adversaries hide their identity. From an insider perspective, attribution might seem easier because within the enterprise network, hosts, software, and the users of those services should be known. A lack of security or detection capabilities could leave blind spots. Split tunneling for web traffic and lack of an always-on VPN solution are just a few

areas where monitoring the behavior of a user can be degraded. On the endpoint, Data Loss Prevention (DLP) and other Endpoint Detection and Response (EDR) agents attempt to bring light to the poorly lit areas of the organization. Often these platforms do not (or cannot) account for encrypted or obfuscated data. In the case of DLP, alerting on encrypted files often yields noise and creates alert fatigue. Active Defense techniques are a great way to reduce alert noise; however, consultation of the legal department is a must before going live.

#### 4.1. Legal Advice

The organization's appetite for implementing Offensive Countermeasures will vary. Before actively engaging any adversaries, both internal and external, an organization should obtain guidance on the limits of the Active Defense techniques. Similar to any information security program, both the legal and management buy-in is a key to success. A simple mechanism for preparing the environment is to review the logon and warning banners for the organization. According to the authors of Offensive Countermeasures, "Warning banners are key because they allow

[the organization] to define the boundaries of [the] networks and the actions [an organization] may take to verify the security of the networks" (Strand et al, 2013). Put another way, warning banners can notify any user, including an insider, that they are closely being monitored for any leaked information, both production and test data. By stating this upfront, the argument of entrapment could be mitigated. Seek legal and management counsel prior to hunting insiders with Offensive Countermeasures. Technology such as the Web Bug Server is a means to hunt the intentional leak of data from an organization.

#### 4.2. The Web Bug Server

The Web Bug Server is essentially a command and control (C2) server for the defender. In its most rudimentary form, the server is a collector for the call back traffic. This server is best

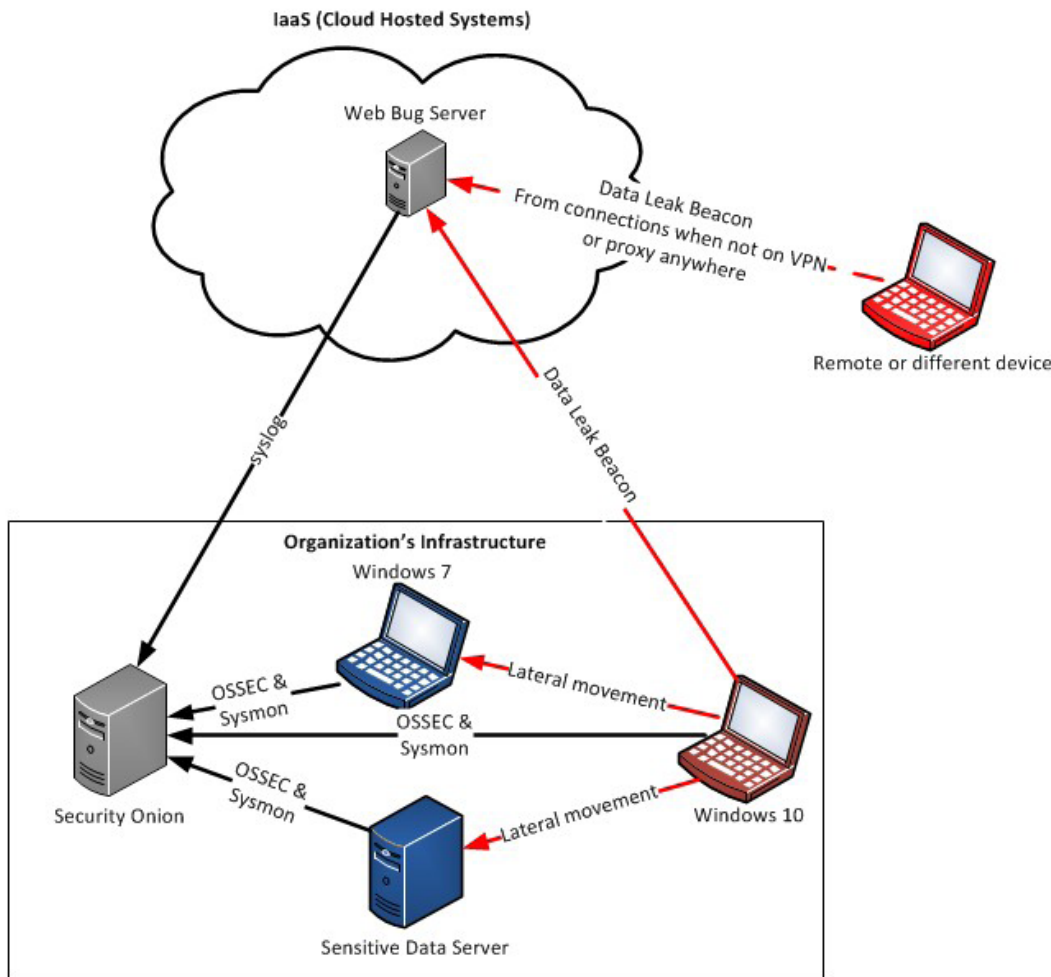


Figure 5: Web Bug Server Conceptual Infrastructure - (Source: Author)

utilized when set up outside of the organization's infrastructure. One example is Amazon's Web Services (AWS), or other Infrastructure as a Service (IaaS) provider. Attributing the server back to the organization could alert the attacker that the document is not only bugged, but being monitored by the organization. The second part to the server is the bugged document itself. This document contains a simple web bug that is not seen by the attacker ("Web Bug Server, n.d.). It can be "embedded inside word processing documents. These bugs are hidden to the casual observer by using things like linked style sheets and 1 pixel images" (Smith, 1999). The important note is that the bug can be placed inside of any document that can process Hyper Text Markup Language (HTML). The primary target file for these bugs would be Office documents, such as .doc, .docx, .xls, .xlsx, and even HTML formatted emails.

Now that both the C2 server and bugged document are in play, the attacker must be enticed with the bugged document. It can be placed in a common share or location the insider might only have access to. Ideally, this share should take effort to access so the argument of accidental disclosure can be lessened. Regardless of how the document makes it out of the organization, when it is opened, a simple callback is sent to the Web Bug Server from the device or host that opens the document. This callback contains identifying information. "Each entry includes the document id which can change by editing the .doc file, the type of media request that was triggered, the IP address the connection came from, and

the time the connection was made" ("Web Bug Server, n.d.). The document ID can be made unique to the user or area it came from, to help with attributing where it came from, or who accessed it. Figure 5 illustrates the conceptual infrastructure for using the Web Bug Server.

This indicator works to suggest that there may be active insiders in an organization. From that knowledge, a more succinct list to identify the insider threat can be formulated.

Although excellent for pinpointing insider threats, the leaked document could also indicate an adversary has made it through the network successfully and achieved his or her actions on the objective. In the case of the leaked document the action or goal is data exfiltration. If the Threat Hunter has a suspicion that there are leaks happening or potentially happening, Mole Hunt helps to narrow the focus.

**4.3. Molehunt**

In some cases, the insiders might already be known, so Molehunt can be used for further attribution. Molehunt takes the simple Web Bug concept to the next level. By leveraging a list, an insider hunt campaign can easily be built by feeding the list to a Python script. Molehunt.py takes the list of insiders and automatically generates unique and bugged documents. Since Molehunt relies on the Web

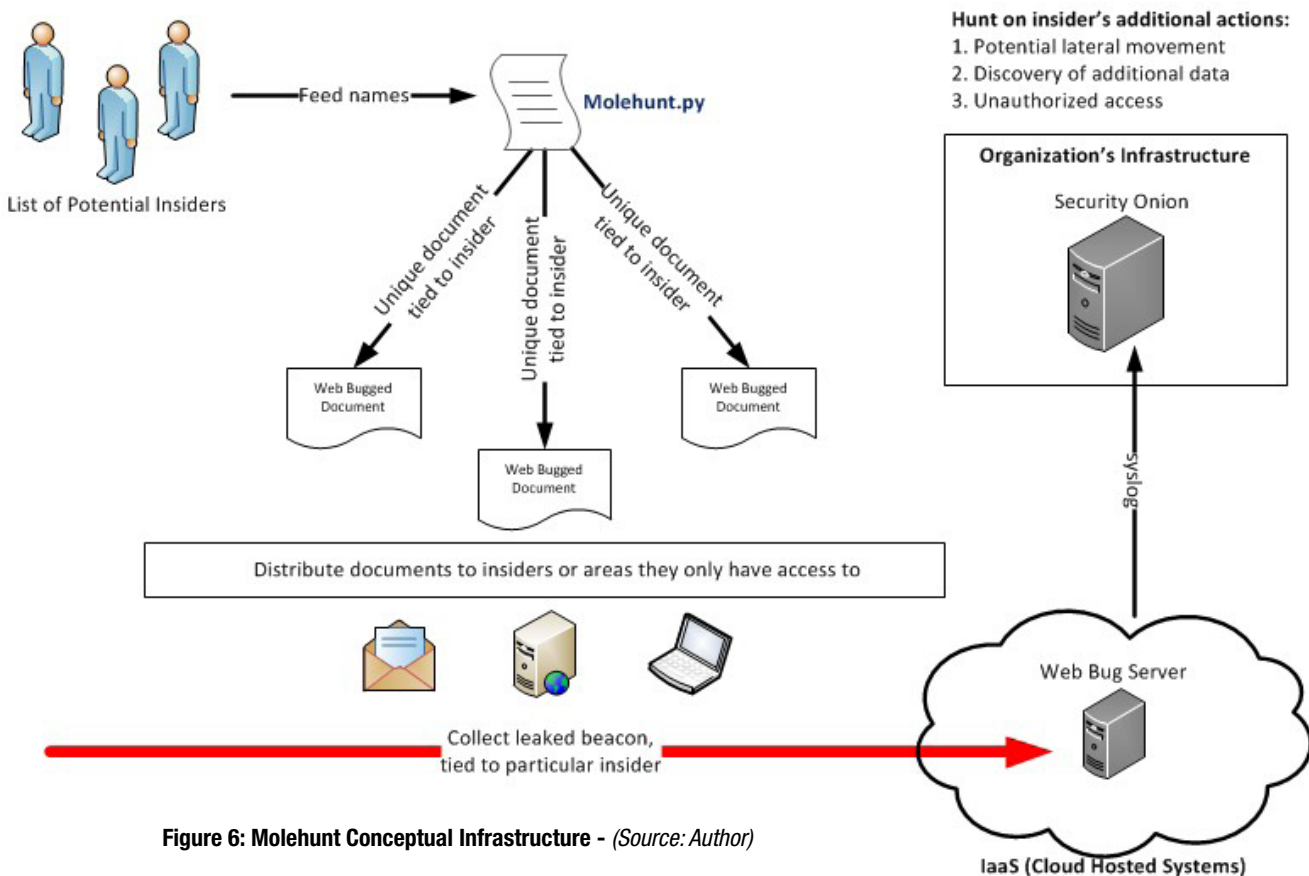


Figure 6: Molehunt Conceptual Infrastructure - (Source: Author)

Bug Server for collecting responses, one can easily dive deeper into the insider hunt, if required (“Molehunt,” n.d.). Figure 6 highlights the features of Molehunt with a recommended configuration.

This data, if received, would be a warning sign that leaks are taking place before any real damage occurs and can even implicate the insider. True to the Threat Hunting definition, this is indeed the proactive pursuit of abnormal, and unwanted, activity on the organization’s systems indicating data exfiltration. The operationalization of Threat Hunting, in particular Active Defense, is the next step in decreasing the detection delta.

### 5. A Threat Hunting Platform: Security Onion

Similar to a rifle or bow, the Threat Hunter requires a set of tools to accomplish the hunt. Commonly thought of as just a Network Security Monitoring (NSM) tool, Security Onion has one of the most expansive sets of security and intrusion detection tools around, including host monitoring. Furthermore, it is open source—free! The core tenants that make Security Onion an extensible platform for Threat Hunting are: full packet capture abilities, network and host –based intrusion detection, built-in analysis tools, and the ability to integrate with the Critical Stack Intel platform for threat feeds (Burks, 2017). All of these, combined with the ability to run in most virtual environments, lend it to being a necessary and vital tool for intrusion detection, both reactive and proactive. The fundamental problem that Security Onion addresses, at least from a Threat Hunting perspective, is the ability to centrally collect log data and network packet captures from nearly anything that can generate a log. New to the platform is the integration of Elasticsearch, Logstash, and Kibana (ELK), which expands the Threat Hunter’s arsenal.

#### 5.1. ELK Hunting

The Elastic Stack is now a feature of Security Onion, which enables the Threat Hunter like never before. From an insider hunting perspective, the alerts received by the Web Bug Server can be forwarded to Security Onion. A guide on how to set this up is located in Appendix C. Once ingested,

the Threat Hunter can leverage Kibana to visualize the data from the leak, as well as, view the context around the systems or users who might be involved. Ultimately, the goal would be to determine if the insider is working alone, with other insiders, or even possibly if an advanced adversary is present and moving laterally. Once the insider or group of insiders has been identified, further hunting activities should be conducted. These activities could start with examining the insider’s lateral movement, enumeration of additional services, or any unauthorized or denied access to data that the user should not be accessing. Preparing the environment ahead of time is a crucial step in the hunting process. Kibana streamlines the searching and analysis of an intrusion, especially when fed with rich data from the organization’s environment.

#### 5.2. Windows Logging and Sysmon

To truly prepare the environment, several areas of logging should be considered, and especially for Windows hosts. In the most basic form, additional auditing for Windows hosts can yield the records required to hunt down human adversaries on an organization’s network. As a more advanced configuration, the introduction of Sysmon, and OSSEC will add even more context to the hunt. Within Security Onion, the means to ingest these logs is built-in. This allows for organizations to more rapidly deploy a comprehensive solution, while maximizing the time the Threat Hunter can spend searching out the human adversaries. Figure 7

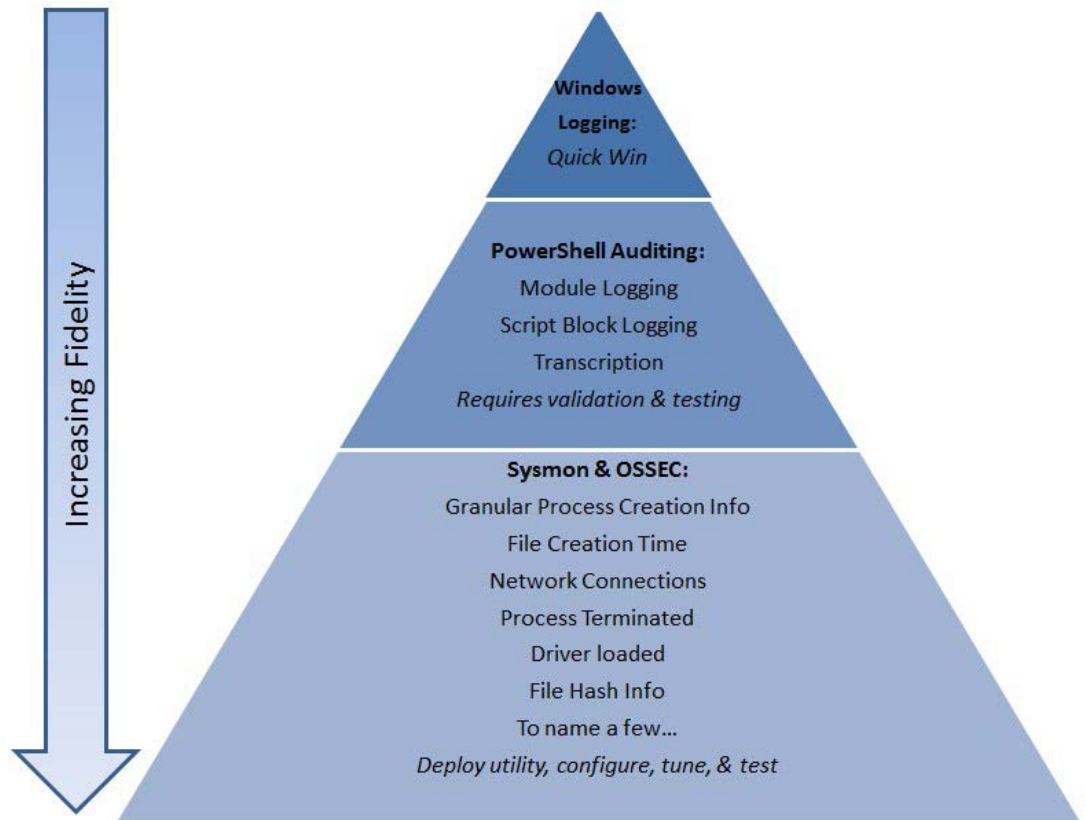


Figure 7: Tiered Top-Down Approach to Enabling Logging for Hunting (“Cheat Sheets,” 2017) - (Source: Author)



depicts a tiered approach to enabling the logging for an enterprise with Hunting in mind.

The recommended log settings can be found in Appendix B. To note, even when logs and network traffic is being analyzed, there is still a possibility that an adversary can fool a system by leveraging a rootkit. Augmenting a platform such as Security Onion with a live memory and disk acquisition capability, such as F-Response, is still recommended. This allows for further analysis by a malware analyst or forensic investigator, if the incident warrants a deeper look. Now that the environment is primed for the hunting season, the adversary's ability to remain undetected is diminished.

## 6. Open Season

With the environment prepped, the focus turns to identifying the active human adversaries. The Web Bug Server and Molehunt will be the primary means used for the active seeding and hunting of the insiders. But how can an organization be so sure that the attacker will go after the bugged documents? The answer might be simpler than expected. The first answer relates to the previously discussed threat modeling and setting the organization's hunt priorities based on the data that requires protection and its relevant adversaries. The second key stems from human nature.

### 6.1. Observed Human Behavior

Both the attackers and victims are fundamentally the same: they are human. When phishing attacks are conducted, the adversary is attempting to exploit the trust of a user.

In many cases, a spoofed website or document is sent to lure the victim into clicking on a link or opening a document. The more authentic the email appears, the more likely the user is to act. From a phishing study of 15 participants, the following was observed: "six Six do not ever click links from email and five will click on a link if it looks interesting. Two only click on links if it from a web-site where they have an account. One will click on links from email only if associated with a specific transaction" (Dhamija, Tygar, & Hearst, 2006). Interestingly, nearly half of the participants would click on an interesting link. Because attackers and phished users are both people, the allure for an adversary to open or exfiltrate interesting data, if that is their intent, is more than likely a motivating factor. For example, if a web server hosted a public directory that contained 50 files and one of those contained a file that was named customer\_data.docx and the rest of the files had a non-descript name like index.html, the likelihood that the customer\_data.docx file would be stolen would be greater.

In a separate study conducted utilizing honey tokens, researchers discovered common motivations for data misuse. The scenario conducted by Shabtai et al. (2016) involved 173 participants who

posed as bankers. Each banker's task was to approve loans by one of two means: The first was to approve the loan legally and the second method was to fund the loan via an outside source, illegally. The more loans and the higher the amount of the loan approved equated to more commission for the banker. Some of the loans were legitimate and some were actually seeded with honeypot. If the loan was approved illegally, the banker risked being fired. What the study uncovered was that "attractive loans (i.e., loans at higher amounts) were more prone to illegal approval" (Shabtai et al., 2016). This means, there was a direct correlation between the amount of personal gain and data misuse. Everyone has their price. The second finding was around religion. "The Religiosity factor was also found to be statistically significant. More specifically, the more religious the participant was, the less illegal actions he or she performed" (Shabtai et al., 2016). Detection was conducted using the honeypots, which enabled the researchers to uncover when decoy data was used and by which banker (Shabtai et al., 2016). Because of the observations of the human behavior, the same tactic can be leveraged against an organization's adversaries when hunting for insiders.

### 6.2. Scenario One: Insiders or Actions on Objectives

The following scenario outlines a very real and probable incident that will be used as an example of hunting an internal, suspected, leaker.

#### Scenario One

- The organization is trying to protect trade secrets
- The organization's biggest threat is the insider (or an advanced adversary already in the network)
- A leak of the information could cause brand degradation and lost revenue
- Data is suspected as being exfiltrated from the organization
- The scope of insider's actions is unknown
- *Hunt the insiders!*

Based on the scenario, it is now up to the Threat Hunter to discover if this activity is occurring on the organization's systems. Included in Appendix E is the flowchart to illustrate the process. To assist with the Hunt, Kibana and attribution via Active Defense techniques will be leveraged.

### 6.3. Discovering Insiders Using Kibana and Active Defense

With the priorities and environment set, the Threat Hunter can begin the pursuit of insiders within the organization's infrastructure. The procedure for this scenario can be referenced in Appendix E. The first phase of the scenario will take the generically bugged document and place it in areas that might either be near truly sensitive data, or in a common area where most users have access to, for example, an

internal departmental share. This location should still take effort to access to diminish any accidental disclosure. The detailed steps on how to bug a .docx file are explained in Appendix C. Additionally, the document could be distributed to the company, or department within the company, via an official looking email. Both of these approaches are the wide net approach. Once seeded, the Hunter can watch for the trap to be tripped. In Figure 8, the alerts received in Security Onion from the Web Bug server are noted.

Based on the alerts, two conclusions can be drawn. The first shows that someone has opened the document. The second inference, and more importantly, is the source IP address. If the IP is the Network Address Translation (NAT'd) IP, it might be assumed the document has not left the organization. In the event the IP is a public IP and not associated with the organization, the document has been opened from outside the organization. If the latter is true, it can be deduced that there is at least one insider leaking information. The question then becomes: has the insider purposely leaked the document or was it a mistake? To track down that answer, a more purpose-built tool can be utilized: Molehunt.

**6.4. Pinpointing the Mole(s)**

With the knowledge that documents are being leaked, the Threat Hunter must determine who the moles are and if the leaks are intentional with the use of Molehunt. Based on the unique document IDs, a list of potential insiders can be determined.

Furthermore, if an administrative or Human Resources (HR) representative can be involved at this stage, they can help in

narrowing the list down. Taking the public source IP from the Web Bug Server alert, a network WHOIS can be utilized to reference the geolocation of the IP. Taking the city list to HR, a list of employees or contractors who live in that area might help with attribution. Although not a perfect method of attribution, the technique removes more uncertainty from who is an insider. Armed with the list of individuals, Molehunt is now ready to accept submissions.

Feeding the list into Molehunt.py will produce uniquely bugged documents for each human. Once created, the Threat Hunter should rename each document to something enticing, while keeping the filename unique, mapped, and referenced so they do not get confused. Appendix C includes the detailed steps of creating the uniquely bugged documents. Distributing the documents is the next challenge. In this round, it is time to place the bugged document into a location that requires the insider a degree of work to access. For example, the analyst should create a directory on a share that the insider would have to actively search to discover. Additionally, an extra warning banner could be placed on the bugged directory, which might seem like a legitimate directory, to further warn the insider. Doing this reduces the case that the insider was ignorant to the fact they were in an area where they should not be. Within that directory, the bugged document, and some others to decrease suspicion, can be staged. Distributing the document will require thought and preparation as to not alert the insider to the fact that they are on a watch list. Now, if the document is exfiltrated, it will have an alert tying directly to the user, or mole. As these alerts are generated, the hope is that the count is small. From there, additional context can be added to the incident.

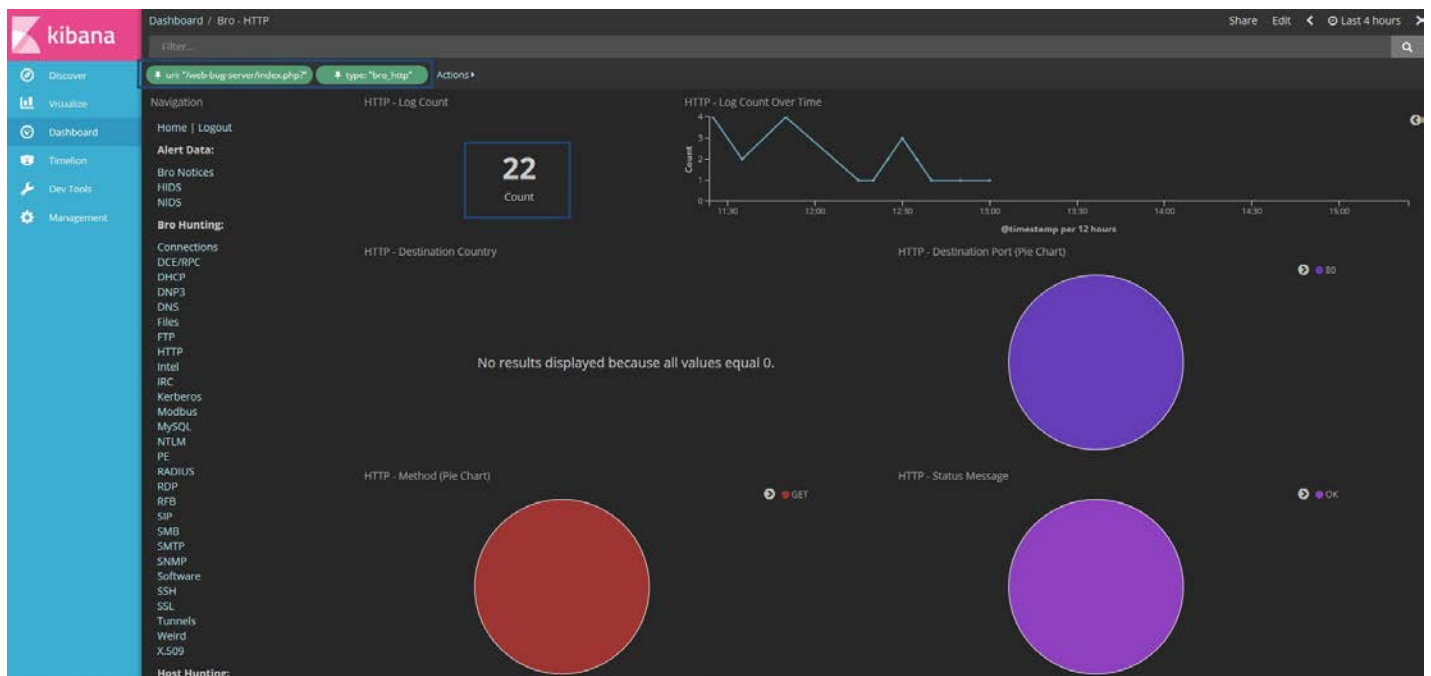


Figure 8: Kibana Dashboard for Viewing the Call Backs to the Web Bug Server - (Source: Author)

The final step in the Hunt is to fully scope the adversary's actions. With a short list of insiders, the Threat Hunter can focus on the additional actions, if any, that were performed. At this point, it is a great hope that the insider has not leaked anything other than the bugged documents. Further Hunting on the actions, such as lateral movement, additional discovery, or additional sensitive data access can be explored. Because the environment is prepped, a historical search into the host logs (event logs, PowerShell, Sysmon, and OSSEC events) can piece the puzzle together. In a recent report by Eduard Kovacs from SecurityWeek, a National Security Agency (NSA) contractor was charged with leaking classified information. The investigation used similar Hunting techniques to hone in on the insider. "An internal audit showed that a total of six individuals had printed the leaked report and one of them was Winner. An analysis of the desk computers used by these six individuals revealed that Winner had contacted the news outlet via email" (Kovacs, 2017). Because the environment was primed, it was a relatively easy process to hunt down the leaker. At the end of the hunting phase, the incident should be scoped and ready to move into the capable hands of the IR Team. In some cases, it might be necessary to understand if the insider is working alone, or in collusion with others— either internal or external. In the case of the NSA contractor, the external communication was identified between her and the Intercept reducing any uncertainty that she was the sole proprietor of the leak (Kovacs, 2017). At this stage in the incident, it might be time to call upon the organization's IR retainer for additional incident handling support.

## 7. Conclusion

Large scale data breaches have occurred and will continue to occur unless the mindsets of security practitioners change. WikiLeaks, the Arab Spring, and the Occupy movements are significant examples of the damage leaked information can do to governments and organizations alike. Bots and machines are not the advanced adversaries, humans are. Because of that reality, Threat Hunting should focus on going after, or hunting, the humans. Simply sifting through logs and alerts may be effective, but it does not lend to a proactive pursuit of intrusions within or against an organization. This is the way it has been done and it produces marginal results, while burning out the human analyst. Although the numbers are decreasing for the identification of a breach, they still lend to a ripe environment for an attacker to succeed. For that reason, Offensive Countermeasures and Threat Hunting must be synonymous. Each organization's appetite for the Active Defense spectrum of AAA will be different. Most can and should focus on the first two A's: Annoyance and Attribution. By determining what needs to be protected and who the adversaries are that the organization faces, lends itself to a strategy or prioritized Hunting program and application of these techniques.

With direction, the Threat Hunter can focus effort and prepare the environment for a successful Hunt. Boiling the ocean will not yield positive results, so an organization might need to start with a platform, such as Security Onion and basic logging. When complete, the next phase can be used to enable further logging, which increases the fidelity of the data a Hunter can analyze. Combined with Active Defense tools of Web Bug Server and Molehunt, the Hunter can go on the offense and proactively seek out insiders who might be leaking data, hopefully before any real data is leaked. Based on the results, Molehunt can help target and validate the moles on an organization's network.

From discovery of a mole, additional context will help to scope the adversary's actions. Based on the organization's needs, this extremely rich data can be used to kick off an IR process or other actions as needed. It is time to let the machines hunt the machines and humans hunt humans. (Merritt & Concannon, 2017).

## REFERENCES

- [1] Adversarial Tactics, Techniques & Common Knowledge. (n.d.). Retrieved June 15, 2017, from [https://attack.mitre.org/wiki/Main\\_Page](https://attack.mitre.org/wiki/Main_Page)
- [2] Apache Logs. (n.d.). Retrieved June 20, 2017, from <https://www.loggly.com/docs/sending-apache-logs/>
- [3] Bachrach, J. (2011, July & aug.). WikiHistory: Did the Leaks Inspire the Arab Spring? Retrieved June 15, 2017, from <http://www.worldaffairsjournal.org/article/wikihistory-did-leaks-inspire-arab-spring>
- [4] Bejtlich, R. (1970, January 01). Try the Critical Stack Intel Client. Retrieved June 20, 2017, from <https://taosecurity.blogspot.com/2015/01/try-critical-stack-intel-client.html>
- [5] Burks, D. (2017, March 16). Introduction to Security Onion. Retrieved June 15, 2017, from <https://github.com/Security-Onion-Solutions/security-onion/wiki/IntroductionToSecurityOnion>
- [6] Carbone, R. (2015, March 19). Using Sysmon to Enrich Security Onion's Host Level Capabilities. Retrieved June 15, 2017, from [https://digital-forensics.sans.org/community/papers/gcfa/sysmon-enrich-security-onions-host-level-capabilities\\_10612](https://digital-forensics.sans.org/community/papers/gcfa/sysmon-enrich-security-onions-host-level-capabilities_10612)
- [7] Cheat Sheets to help you in configuring your systems. (2017). Retrieved June 15, 2017, from <https://www.malwarearchaeology.com/cheat-sheets/>
- [8] Chuvakin, A. (2017, April 06). My "How to Hunt for Security Threats" Paper Published. Retrieved June 15, 2017, from <http://blogs.gartner.com/anton-chuvakin/2017/04/06/my-how-to-hunt-for-security-threats-paper-published/>
- [9] Costa, D. L., Michael, A. J., Matthew, C. L., Perl, S. J., Silowash, G. J., & Spooner, D. L. (2016, May). An Insider Threat Indicator Ontology. Retrieved July 05, 2017, from [https://resources.sei.cmu.edu/asset\\_files/TechnicalReport/2016\\_005\\_001\\_454627.pdf](https://resources.sei.cmu.edu/asset_files/TechnicalReport/2016_005_001_454627.pdf)
- [10] Dhamija, R., Tygar, J. D., & Hearst, M. (2006, April). Why Phishing Works. Retrieved June 15, 2017, from [http://people.ischool.berkeley.edu/~tygar/papers/Phishing/why\\_phishing\\_works.p df](http://people.ischool.berkeley.edu/~tygar/papers/Phishing/why_phishing_works.p df)
- [11] Gregory, P. H. (2017, April). Threat Hunting for Dummies. Retrieved June 15, 2017, from <https://www.carbonblack.com/resource/threat-hunting-dummies/>

[12] Kovacs, E. (2017, June 06). NSA Contractor Charged With Leaking Russia Hacking Report. Retrieved July 06, 2017, from <http://www.securityweek.com/nsa-contractor-charged-leaking-russia-hacking-report>

[13] Lee, R. (2016, November 28). FOR508 - Advanced Incident Response and Threat Hunting Course Updates: Hunting Guide. Retrieved June 15, 2017, from <https://www.youtube.com/watch?v=C-0JD1Fwk7U>

[14] Lee, R. M., & Bianco, D. (2016, August). Generating Hypotheses for Successful Threat Hunting. Retrieved June 15, 2017, from <https://www.sans.org/reading-room/whitepapers/threats/generating-hypotheses-successful-threat-hunting-37172>

[15] Merritt, K., & Concannon, B. (2017, May 16). Vector8 Threat Hunting & Advanced Analytics Course [PDF]. Denver.

[16] Molehunt. (n.d.). Retrieved June 15, 2017, from <https://github.com/adhdproject/adhdproject.github.io/blob/master/Tools/Molehunt.md>

[17] M-Trends 2017 A View From the Front Lines [PDF]. (2017). California.

[18] Robish, E. (2016, November 7). Bugging Microsoft Files: Part 1 – Docx Files using Microsoft Word. Retrieved June 20, 2017, from <https://www.blackhillinfosec.com/?p=5409>

[19] Romero, F. (2010, November 29). Top 10 Leaks. Retrieved June 15, 2017, from [http://content.time.com/time/specials/packages/article/0,28804,2006558\\_2006562\\_2006567,00.html](http://content.time.com/time/specials/packages/article/0,28804,2006558_2006562_2006567,00.html)

[20] Saba, M. (2011, September 17). Wall Street protesters inspired by Arab Spring movement. Retrieved June 15, 2017, from <http://www.cnn.com/2011/09/16/tech/social-media/twitter-occupy-wall-street/>

[21] Shabtai, A., Bercovitch, M., Rokach, L., Gal, Y., Elovici, Y., & Shmueli, E. (2016).

[22] Behavioral Study of Users When Interacting with Active Honeytokens. ACM Transactions On Information & System Security (TISSEC), 18(3), 9:1-9:21. doi:10.1145/2854152

[23] Smith, R. (1999, November 11). The Web Bug FAQ. Retrieved June 15, 2017, from [https://w2.eff.org/Privacy/Marketing/web\\_bug.html](https://w2.eff.org/Privacy/Marketing/web_bug.html)

[24] Strand, J., Asadoorian, P., Robish, E., & Donnelly, B. (2013). Offensive countermeasures: the art of active defense. Place of publication not identified: Publisher not identified.

[25] Sundaramurthy, S. C., McHugh, J., Rajagopalan, X., & Wesch, M. (2014, Sept. & oct.). An Anthropological Approach to Studying CSIRTs. Retrieved June 15, 2017, from <https://pdfs.semanticscholar.org/d31a/1c631d9a74f144c5291ed50765ac36e760ad.pdf>

[26] Web Bug Server. (n.d.). Retrieved June 15, 2017, from <https://github.com/adhdproject/adhdproject.github.io/blob/master/Tools/WebBugServer.md>

[27] WikiLeaks Release 1.0. (2009, December). Retrieved June 15, 2017, from [https://archive.org/details/26c3-3567-en-wikileaks\\_release\\_10](https://archive.org/details/26c3-3567-en-wikileaks_release_10)

## ABOUT THE AUTHOR

**Matt Hosburgh** is a passionate security practitioner, currently working as a Cyber-Threat Hunter for a Philadelphia based company. With over 15 years of experience working in various security disciplines, Matt began his InfoSec career while serving in the U.S. military.

After the Marine Corps, he transitioned from his military role to work as a Senior Security Analyst for United States Citizenship and Immigration Services (USCIS). During his time at USCIS, he was an integral part of the Security and Network Operation Center (SNOC) and the Computer Security Incident Response Team (CSIRT). Following that responsibility, Matt was the Senior Security Engineer for a mid-stream oil and gas company where he supported the company in securing both IT and Operational Technology (OT) systems.

Matt holds a Master's of Science in Information Security Engineering from the SANS Technology Institute, and maintains several GIAC Certifications, to include the GSE.

LIKE AND FOLLOW US  
ON SOCIAL MEDIA!




Search:  



# R&E Gateway

Powered by DTIC

<https://www.dtic.mil>



Propel your research, gain new insights  
and bring to life your warfighter  
technology concepts and solutions.

- *Over 4 Million Technical Reports*
- *DoD Research Projects*
- *DoD-Published R&E Journal*
- *Planned Research*
- *24x7 Virtual Workspace*



Get started at

<https://go.usa.gov/xQ33R>



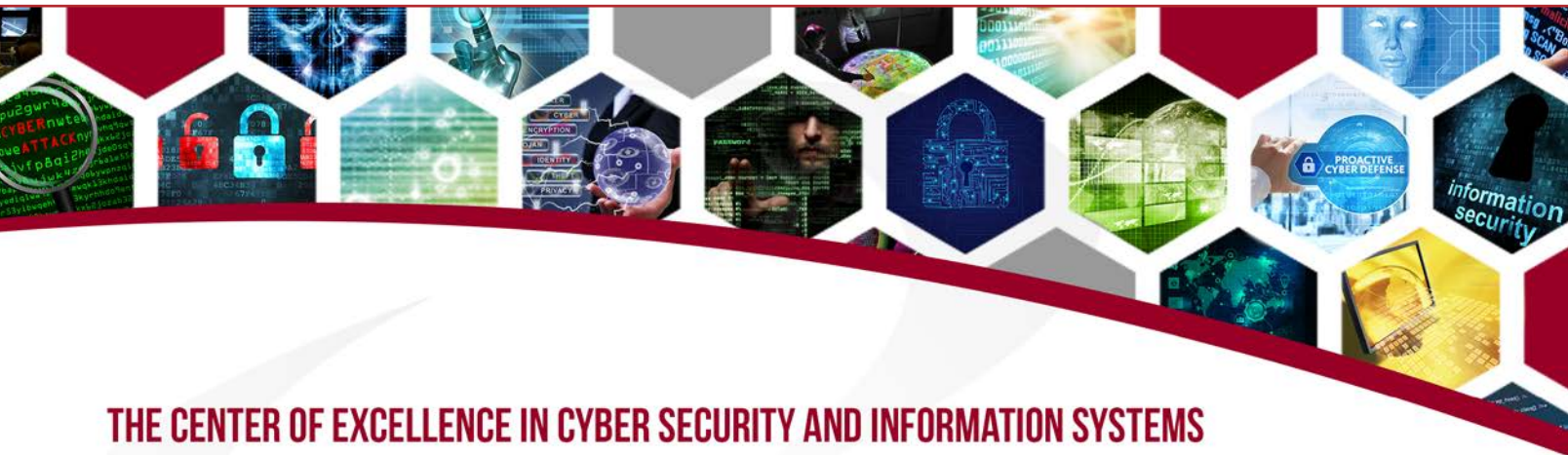
The Defense Technical Information Center (DTIC) is DoD's authoritative source for scientific and technical (S&T) information! For more information on DTIC contact **1-800-225-DTIC (3842)**, and choose **option 1** or email: [dtic.belvoir.us.mbx.reference@mail.mil](mailto:dtic.belvoir.us.mbx.reference@mail.mil)

**Cyber Security and Information Systems  
Information Analysis Center**

266 Genesee Street  
Utica, NY 13502



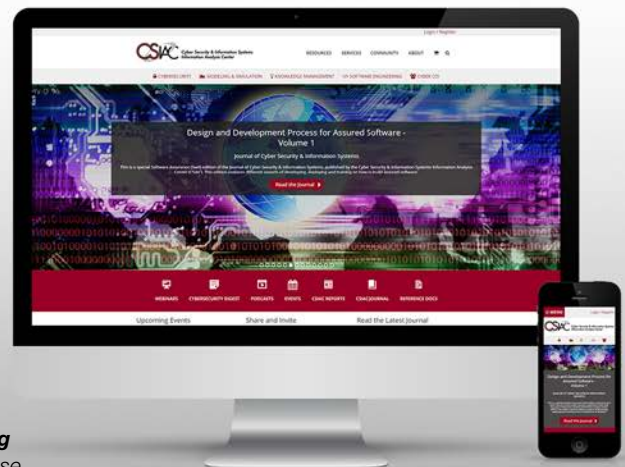
Address Service Requested



## THE CENTER OF EXCELLENCE IN CYBER SECURITY AND INFORMATION SYSTEMS

*Leveraging the best practices and expertise from government, industry, and academia in order to solve your scientific and technical problems*

<https://www.csiac.org/journal/>



To unsubscribe from CSIAAC Journal Mailings please email us at [info@csiac.org](mailto:info@csiac.org) and request that your address be removed from our distribution mailing database.