Spring 2011

Towards A Framework For Maintaining Defensibility In Encrypted Network Environments

John Prewett
Regis University

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

TOWARDS A FRAMEWORK FOR MAINTAINING DEFENSIBILTY IN ENCYPTED NETWORK ENVIRONMENTS

A THESIS

SUBMITTED 6, May, 2011

TO THE DEPARTMENT OF INFORMATION TECHNOLOGY

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

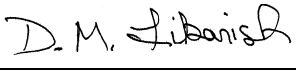
OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN INFORMATION ASSURANCE

BY

John Prewett

APPROVALS



Daniel Likarish, Thesis Advisor

Shari Plantz-Masters

Robert T. Mason

Abstract

Network security professionals improve confidentiality and integrity of information technology resources when they incorporate encryption schemes into the transmission of network packets across their respective infrastructures. Ironically, network engineers and administrators that incorporate encryption strategies across their infrastructures must simultaneously confront the limitations of end-to-end encrypted network packets inasmuch as they severely impair visible, defensible network architectures. This project demonstrates how security professionals charged with maintaining network visibility can deploy encryption across their topologies without fear of compromising their ability to capture – then fully analyze – network traffic. In so doing, information technology industry practitioners and researchers may confidently move forward with the task of maturing a framework for maintaining defensibility in encrypted network environments.

Acknowledgements

The completion of this thesis project would have been exponentially more difficult without the unyielding support and encouragement I received from my wife, Megan Prewett. I also appreciate insight and feedback received from course instructors, academic colleagues, and industry practitioners throughout the duration of my graduate-student tenure at Regis University.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

> Framework \frām-, wərk\: a simplified description of a complex entity or process.
>
> *www.websters-dictionary.org*

Value and Importance of Frameworks

Thoughtfully designed frameworks reduce complexity. Business practitioners and research professionals alike create, analyze, refine, and reuse frameworks for the purpose of clarifying otherwise obscure or unwieldy activities. For example, IBM business systems planner John Zachman (1987) revolutionized the modern corporate landscape when he proposed what matured into a widely adopted (and often emulated) framework for the effective and efficient integration of information technology (IT) assets into day-to-day business operations. Similarly, contemporary software engineers the world over rely on Agile, Waterfall, or Spiral frameworks (to name a few) in the process of designing and producing highly complex yet reliable software and database applications that government agencies, private businesses, and individual consumers find indispensable (Ambler, n.d. & Elucidata, n.d.).

Nowhere do frameworks prove their value more obviously than to those professionals charged with architecting, implementing, and maintaining complex technology systems. As Jeanne Ross (2004 & 2005) concluded, the process of reducing complexity not only saves time and money, it also results in improved competitive advantage for those willing to understand and practice the nuances of industry-applicable frameworks. In the case of IT, properly implemented frameworks further minimize unnecessary expenditures, reveal flaws in design assumptions,

improve operational efficiency, identify points of potential failure, and mitigate future risk (Bernard, 2005).

Value and Importance of Network Encryption

Encryption serves as a fundamental cornerstone of computer network security (Pfleeger & Pfleeger, 2007). Thoroughly engineered encryption schemes provide confidentiality and integrity of data packets traversing both wired and wireless network topologies. Without robust encryption algorithms, such modern services as on-line banking, electronic commerce, and remote telecommunications would all but cease to exist.

While encryption supports a multitude of important activities 21st-century technology users now find indispensable, nefarious individuals and/or criminal syndicates can easily employ the same encryption methodologies originally intended to fuel global economies of the future to (instead) initiate, perpetuate, and obfuscate their own movements and activities from even the most vigilant crime fighters. It logically follows that if legitimate financial institutions can (and do) use encryption to protect millions of legitimate transactions totaling trillions of dollars, technologically inclined thieves can (and do) also use the same encryption strategies to hide their own illicit initiatives without raising even the slightest real-time suspicions. In this respect, the value and benefits originally associated with encryption quickly become liabilities that have the very real potential of severely harming individual consumers, business organizations, government agencies, and peace-keeping operations around the world.

Problem Statement – The Encryption Dichotomy

Security professionals charged with protecting corporate infrastructures, customer information, business partner relationships, and/or national secrets can ill-afford to ignore or minimize the important role encryption plays in securing both logical and physical digital assets. Because of its inherent value to security architectures of every size and configuration, encryption will continue to sustain the core activities of modern economies far into the foreseeable future.

Nonetheless, technology professionals must simultaneously acknowledge that end-to-end data encryption across their respective topologies constitutes a serious problem primarily because end-to-end data encryption necessarily undermines network security. Richard Bejtlich (2005) asserted that the concept of *defensibility* – where network engineers and administrators design and maintain network topologies best suited to resist unauthorized intrusions – most appropriately defines comprehensive computer network *security.* Bejtlich further elaborated that defensible computer networks must easily facilitate *visibility* or the ability for **authorized personnel** to **meaningfully monitor all data traffic** that traverses a given network topology.

That *visibility* leads to *defensibility,* which finally leads to *security,* accentuates a fundamental problem with any end-to-end data encryption methodology: end-to-end data encryption severely constrains attempts on the parts of authorized personnel to meaningfully inspect and analyze network traffic (Bejtlich, 2005). Nowhere does the dichotomy of encryption's inherent benefits and liabilities more critically apply than to the authorized inspection and analysis of network traffic generated by unauthorized network users. If unauthorized intruders use robust encryption schemes to obscure their movements and activities, even the most sophisticated and rigorous network monitoring strategies will prove wholly ineffective.

A Framework for Resolving the Encryption Dichotomy

Network security engineers and administrators that successfully capitalize on the benefits of data encryption (i.e., increased confidentiality and integrity) while simultaneously minimizing its concurrent risks (i.e., decreased visibility and defensibility) stand to best thwart attempts of unauthorized intrusion and subsequent ex-filtration of proprietary information. Instead of limping through the network security landscape with an Achille's heel, security professionals that resolve the encryption dichotomy brandish double-edge swords that prove that much more effective at securing digital assets.

Unfortunately, network security professionals sincere about resolving the encryption dichotomy have very few resources at their disposal when trying to implement the most secure yet visible encryption architectures across their topologies. To be sure, an abundance of books, journal articles, and on-line resources explain the mathematic principles behind encryption, detail specific encryption algorithms and associated network protocols, or outline design principles of secure computer networks, but no framework – no simplified description of an otherwise complex process – exists upon which network engineers and administrators may rely to maintain visibility in their encrypted network environments.

Significance

A framework formulated to improve network defensibility through full-content analysis of encrypted network traffic would prove invaluable. Understanding and implementing the subtleties of contemporary encryption algorithms based on complex mathematic operations can prove challenging enough. Understanding how to integrate the same complex operations across an enterprise **while also maintaining visibility** (and, therefore, improving defensibility and

security) can prove more challenging still. A thoughtfully designed framework has the very real potential of minimizing such challenges and takes significant strides towards resolving the encryption dichotomy.

Importantly, the significance of such a framework extends beyond the mere simplification of an otherwise complex challenge. In addition to improving enterprise-wide security, a framework for maintaining visibility in encrypted network environments carries with it all the implied benefits generally associated with framework implementation (e.g., improved design, decreased waste, mitigated risk, etc.). Moreover, such a framework stands to improve the competitive advantage and market position of those organizations willing to adopt and practice said framework.

Project Objective and Limitations

Successful analysis of encrypted network traffic ultimately requires knowledge of and access to the software keys originally employed in the process of converting clear-text (i.e., easily discernible and understandable) data into cypher-text (i.e. obscure and incomprehensible) data (see Chapter 3). Therefore, a worthwhile framework dedicated to resolving the encryption dichotomy must (at a minimum) adequately address encryption key storage and retrieval.

This thesis project makes a contribution towards a forthcoming encryption/decryption framework by analyzing critical hardware and software encryption components commonly deployed across network topologies for the purpose of determining the degree to which they support encryption key storage and retrieval methods. As such, the eventual development of a framework for maintaining defensibility across encrypted network environments begins by answering the following research questions: will analysis of critical hardware and software

encryption components commonly deployed across network topologies support the host-to-host decryption process thereby demonstrating practical the eventual development of a framework for maintaining defensibility across encrypted network environments?

This thesis project intends to formulate the *beginnings* of a working, viable encryption framework upon which the security community may confidently rely as the IT industry maneuvers towards reaping the rewards of encryption while simultaneously addressing inherent risks associated with the very same. However, developing and publishing an exhaustive framework that comprehensively resolves the encryption dichotomy will require extensive future research investment from private businesses, government agencies, and academic institutions. Although this project marks an initial step in the direction of creating a much-needed framework, it does *not* result in a finalized working framework. The resolution of a full-functioning framework model falls to research practitioners representing a variety of market sectors. Chapter 6 of this paper proposes topics for future research that have the potential of contributing towards the maturation of an encryption/decryption framework.

Chapter 2 – Literature Review

Developments in Network Encryption Research

Because of encryption's pivotal role in computer network security architectures, industry practitioners, academic researchers, technology companies, and government agencies representing a variety of skills and experiences have published copious volumes of information dedicated to topics ranging from fundamental encryption mechanics (Lewand, 2000) to complex trust models based on encrypted authorization (Liu, 2008). Forouzan (2008) and Burnett & Paine (2004) focused their attentions differentiating between symmetric and asymmetric bloc cyphers and outlining encryption-based network protocols like IPsec and SSL. The System Administration, Networking, and Security (SANS) Institute (Forward, 2002 & Oxenhandler, 2003) and National Institute of Standards and Technology (Frankel, 2010) both suggested deployment strategies for encryption methodologies across small-scale and enterprise-wide computer networks, while Schneier (1996) provided detailed instructions for software engineers charged with integrating encryption algorithms into their computer programs.

Although network security professionals find value in each of the above technical resources, they serve little use for those individuals and teams of specialists responsible for maintaining visible computer network topologies predominated by end-to-end encryption protocols. Both Mackey (2003) and Ciampia (2009) eluded to critical network encryption design features that have the potential of proving useful in visible, encrypted network environments, but they failed to provide a blueprint so others could implement their advice. Even Bejtlich (2005, p. 618) – an ardent supporter of visible, defensible, secure network – acknowledged encrypted

network packets have the potential of thwarting network forensics investigations, yet he stopped short of articulating a working resolution to the encryption dichotomy.

Within the past half-decade, a few academics have made indirect contributions that could indirectly benefit a framework for maintaining visibility in encrypted network environments. Wright (2006) and Gebski (2006) recommended inferencing techniques and protocol signature identification to ascertain the *intent* of electronic messages. In a similar fashion, Koch (2010) proposed command sequence analysis combined with probability algorithms as a method for hypothesizing (then acting upon) network communications packets assumed to carry malicious payloads.

Genuine though the intentions of the above approaches may be, they fail to consider storage and retrieval of encryption keys, which – by extension – predicates **meaningful full-content data analysis** of computer network traffic. Without visibility of the entire, unadulterated contents of any given network packet, network security professionals must rely on best-effort (i.e., best-guess) strategies for thwarting attacks against their infrastructures. While best-effort strategies certainly have their place within the IT community, they prove counter-productive to organizations defending their courses of action in legal proceedings that place higher price tags on verifiable actions rather than assumptions of intent.

Review of available resources dedicated to modern computer encryption techniques and their applicability to network security reveals a fundamental deficiency: network security engineers and administrators lack even a basic framework for integrating end-to-end data encryption into their respective network topologies that simultaneously supports the critical ability to meaningfully perform as-needed, full-content analysis of encrypted data payloads. This thesis project intends to make a contribution towards resolving this deficiency.

Chapter 3 – Decryption = Algorithm + Keys

The Unhappy Marriage of Encryption and Defensible Networks

In order to preserve network visibility (and, by extension, network defensibility), security administrators must maintain the ability to dissect network packets within their respective topologies and meaningfully ascertain their individual payloads. Of course, network traffic transmitted as clear-text (i.e., without encryption) presents very little challenge to security personnel with access to multiple capture locations and software tools (e.g., Wireshark) brilliantly engineered to capture and parse network packet content.

Unfortunately, neither the most efficiently designed network topologies nor the most sophisticated forensics tools have any practical use when trying to dissect fully encrypted data packets transmitted and received by individual workstations and/or servers. (Note: This condition has everything to do with access to encryption keys needed to decrypt network packets and will garnish detailed attention in the following sections of this chapter.) In these situations, encrypted network traffic looks like nothing more than random, nonsensical characters that necessarily prevent meaningful interpretation. Table 1 (next page) compares an unencrypted network message to its encrypted counterpart and further illustrates the burden of trying to meaningfully interpret encrypted messages intercepted as they traverse network topologies.

Unencrypted Network Message (also known as *clear text*)		**Same Network Message Encrypted** (also known as *cypher text*)
Start	→	Finish
I'm ready to install computer viruses!		bb9ca9aa479de85cb80397ebe29742f67163182e 01941f1c05b59a4469632c6ecc869012ba3d0462

Table 1: Comparison of a clear-text message against the same message after encryption using DES, a well-publicized and commonly employed encryption algorithm. See *Appendix A* for a detailed explanation of references and steps used to convert the above clear-text message into its encrypted, cypher-text counterpart.

Using Table 1 (accompanied by Appendix A) as a simple yet accurate working example, the profound implications of encrypted traffic for defensible network infrastructures become glaringly obvious: fully executed host-to-host encryption algorithms scramble network packet payloads to such degrees that security administrators loose practical visibility into the traffic that traverses their organizations' network backbones and associated trunks. Lack of visibility has very few negative implications in trusted environments where all users behave as they should, but lack of visibility proves disastrous in environments where unscrupulous computer and network hackers so much as intend to lurk.

Importance of Encryption Keys – An Encryption Primer

Encryption algorithms perform mathematic operations on clear-text data to the point where the clear-text data becomes unrecognizable cypher text (as exemplified in Table 1). After encryption at the point of origin, computers and cooperating network devices transmit cypher text to a destination (usually another computer) that then must employ the original encryption algorithm to unscramble the cypher text into discernible clear-text messages (Burnett & Paine, 2001).

However, encryption algorithms themselves do *not* insure confidentiality and integrity of scrambled messages. After all, encryption algorithms are well documented, and anyone willing

to invest a little research energy can learn critical mathematic operations performed by a given encryption algorithm, then use the information learned to decode any and all cypher text generated through use of the algorithm(s) in question.

The general availability and access to encryption algorithms necessitates that the overall success of encryption depends on a secret variable that encryption algorithms include in their otherwise ubiquitously publicized mathematic operations. This secret variable – known as an encryption key – helps perform the calculations that ultimately result in cypher text.

As applied to the malicious network message introduced in Table 1, an XOR operation (see Appendix B) of the original clear-text message against a predefined encryption key resulted in a fully encrypted network message. Table 2 (below) more accurately depicts the encryption process outlined in Table 1, particularly because Table 2 includes the working encryption key that ultimately resulted in the unintelligible cypher text introduced in Table 1.

Unencrypted Network Message (also known as *clear text*)		**Encryption Key**		**Same Network Message Encrypted** (also known as *cypher text*)
Start	→	XOR operation	→	Finish
I'm ready to install computer viruses!		3b3898371520f75e		bb9ca9aa479de85cb80397ebe29742f67163182e01941f1c05b59a4469632c6ecc869012ba3d0462

Table 2: Encryption algorithms require encryption keys to convert clear-text messages into cypher text. See *Appendix A* for a detailed explanation of references and steps used to convert the above clear-text message into its encrypted, cypher-text counterpart.

Encryption only works when both (or all) parties involved in the transmission and receipt of cypher text have access to the encryption algorithm **and** encryption key(s) used to scramble the original clear-text message. Absence of either the algorithm or the key(s) at the endpoint receiving electronic messages – or the collection point used to capture and record network traffic – results in worthless messages, primarily because the receiving party (or capturing party, in the

case of network surveillance) cannot properly reverse the XOR process and decode the transmitted cypher text into something meaningful.

Referencing the sample clear-text network message introduced in Table 1, reversing the XOR operation of the cypher text message against the exact same encryption keys decodes the encrypted network packet into something meaningful and clearly reveals the malicious intent of the cypher-text message (see Table 3).

Encrypted Network Message (also known as *cypher text*)	**Encryption Key** **(NOTE: Same key and mathematic operation used in Table 2)**	**Same Network Message Decrypted** (also known as *clear text*)
Start →	XOR operation →	Finish
bb9ca9aa479de85cb80397ebe29742f67163182e01941f1c05b59a4469632c6ecc869012ba3d0462	3b3898371520f75e	I'm ready to install computer viruses!

Table 3: Encryption only works when both operation(s) and key(s) originally used to encrypt the message are also used to decrypt the message.

However, alterations to any (or all) encryption keys – or applying a different mathematic operation (e.g., AND instead of XOR) to the decoding process – necessarily results in messages that severely hinder meaningful interpretation with as much frustration as the original encrypted message. Table 4 (next page) simply yet accurately illustrates the wholly ineffective outcome of the decryption process using an altered encryption key. Likewise, Table 5 (next page) simply yet accurately illustrates the wholly ineffective outcome of the decryption process using an AND operation instead of an XOR operation.

Encrypted Network Message (also known as *cypher text*)	**Altered Encryption Key; Same Operation**	**Same Network Message Decrypted** (also known as *clear text*)
Start →	XOR operation →	Finish
bb9ca9aa479de85cb80397ebe29742f67163182e01941f1c05b59a4469632c6ecc869012ba3d0462	e57f0251738983b3	¬X_q#ìâ×□_v>SàíÀ□#b1SêìÞ□v4#©õÚ□q4□¨£□

Table 4: During the decryption process, reliance on a key different than the key originally used to scramble the clear-text message results in an equally indiscernible final message.

Encrypted Network Message (also known as *cypher text*)	**Same Encryption Key; Different Operation**	**Same Network Message Decrypted** (also known as *clear text*)
Start →	AND operation →	Finish
bb9ca9aa479de85cb80397ebe29742f67163182e01941f1c05b59a4469632c6ecc869012ba3d0462	3b3898371520f75e	□#□####C□##p##e##□#□##tÃ###Q##d□###Q##□□

Table 5: Similarly, during the decryption process, reliance on a mathematic operation different than the mathematic operation originally used to scramble the clear-text message results in an equally indiscernible final message.

Ineffective Network Surveillance in Encrypted Environments

Importantly, this basic deciphering formula (decryption = algorithm + keys) holds true even for legitimate, trustworthy security administrators charged with the responsibility of maintaining visible, defensible, and secure computer networks. If security professionals engaged in network surveillance fail to correctly identify either (or both) the encryption algorithm(s) and encryption key(s) originally used to scramble network traffic, their efforts will prove utterly ineffective and wholly benign (as exhibited in Tables 4 and 5 above).

It's equally worth noting that failure on the part of security professionals to decipher encrypted network messages that traverse network topologies (due to inaccurate identification of either the encryption algorithm and/or encryption keys) does *not* render network messages ineffectual or less potent once they arrive at their destination. If a destination computer receiving encrypted network packets employs the same encryption algorithm and encryption keys used by

the sending computer to encrypt the original message, the receiving computer will correctly decrypt the network packet and act upon its payload, despite the fact that security administrators successfully captured (but failed to decipher) the malicious message as it negotiated its way across the network topology. The fact remains, an encrypted computer virus is still a computer virus that will unleash havoc once transmitted, decrypted, and then executed at its final destination.

Figure 1 (next page) depicts a completely *ineffective* network surveillance scenario. In Figure 1, a network security administrator successfully captured the encrypted network packet introduced in Table 1. However, the network security administrator possessed an encryption key different than the key used by the sending computer to originally scramble the transmitted network message. As such, the security professional falls into a condition best exemplified by Table 4 because he/she cannot properly decode the network message. Conversely, the receiving computer possessed both the encryption algorithm and encryption key used by the sending computer to scramble the network message; consequently, the receiving computer properly decoded the network message originally transmitted by the sending computer. In the simple scenario illustrated in Figure 1, the security professional monitoring network traffic had precious little information to guide his/her next steps in defending technology assets while the receiving computer clearly understood what malicious actions follow.

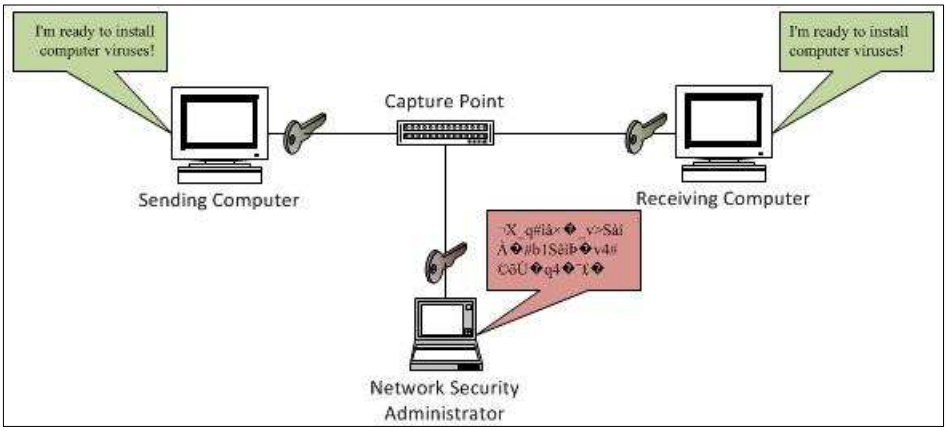


Figure 1: An ineffective network surveillance scenario in which a network security administrator cannot successfully decode an encrypted network message. In the illustration above, the network message happens to be malicious, but the network security administrator has no defensive recourse because he/she cannot meaningfully ascertain the message's intent.

Effective Network Surveillance in Encrypted Environments

This chapter, with the inclusion of explanations, tables, and illustrations, merely serves to establish that visibility, defensibility, and security of physical and logical IT assets in encrypted network environments hinges on the ability of network defense practitioners to properly collect, manage, and apply the original encryption keys used to convert clear-text messages into cypher text that eventually propagates through a given topology. With access to the original encryption keys, network security professionals can meaningfully decrypt encrypted network packets, quickly identify malicious messages of all varieties and, more importantly, take proactive steps to thwart the execution of malicious code as illustrated in Figure 2 (next page).

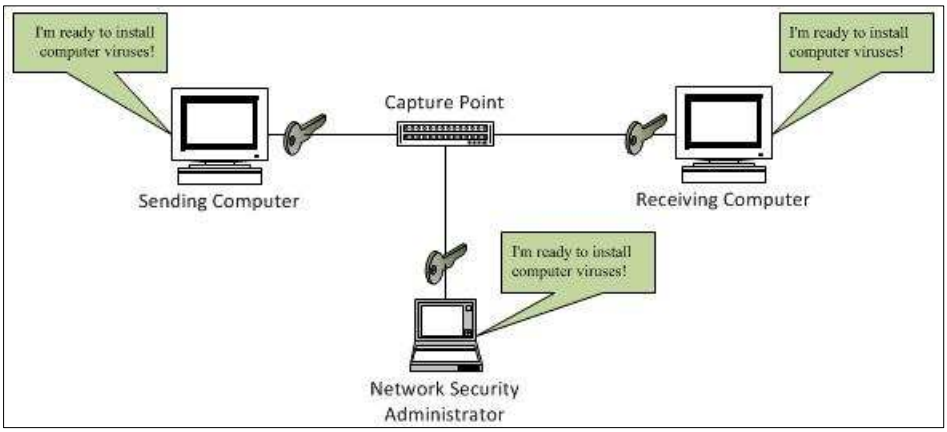


Figure 2: An effective network surveillance scenario in which a network security administrator successfully decoded a network message. In the above illustration, the network security administrator can now appropriately respond to the malicious action uncovered as a result of having properly deciphered the network message.

Because encryption keys play such pivotal roles in maintaining defensible network infrastructures, any contribution to formulating a viable framework that resolves the encryption dichotomy must first identify how network security professionals exercise their ability to collect and correctly apply encryption keys for the purpose of meaningfully decoding encrypted network packets. By analyzing critical hardware and software encryption components commonly deployed across computer network topologies, this thesis project will explore the possibility of network packet encryption key retrieval, the success of which constitutes a critical cornerstones upon which a future, full-functioning framework for resolving the encryption dichotomy will emerge.

Chapter 4 – Research Methodology

Action Research

As a general concept, researchers rely on research methodologies to systematize their procedures for generating (or observing), collecting, interpreting, and finalizing data considered sufficiently necessary to answer their respective research questions (Leedy, 2005). While convention allows for the adoption of more than one methodology per research project, the defining principles and characteristics of action research (AR) most appropriately applied to this initial contribution towards a framework for maintaining defensibility in encrypted network environments.

In outlining the nuances of AR, Richard Baskerville (1999) explained that, by definition, AR research projects include three critical steps: 1) action planning, 2) action taking, and 3) evaluating. In the *action planning* step, researchers develop working prototypes for the expressed purpose of solving problems under investigation. *Action taking* effectively requires researchers to deploy their prototypes and collect resulting data, and *evaluating* simply involves analyzing data generated by prototypes to determine the degree to which they answer research questions. Importantly, Baskerville (p. 17) concluded that AR practitioners could apply results (both positive and negative) culminating from the action planning, action taking, and evaluating steps to guide the formulation and revision of theoretical frameworks, an envisioned final goal of this research project.

Alistar Cockburn (2003, p. 14) prescribed action research (AR) methodology for those researchers intent on "[improving] practitioners' practice". Because a framework for maintaining defensibility in encrypted network environments has the very real potential of improving the processes and procedures of network engineers and administrators engaged in the practice of maintaining visible computer network topologies that also include host-to-host encryption techniques, AR even more appropriately applied to this thesis project and, therefore, served as a substantive guide for its completion.

Action Planning – Prototype Creation

Consistent with AR methodology, formulating a viable framework that resolves the encryption dichotomy first requires prototyping a lab environment that includes 1) network nodes (i.e., computers) for originating and receiving network traffic, 2) an encryption scheme for converting clear-text network packets into cypher text, 3) a collection node that captures encrypted traffic, and 4) techniques for reversing the encryption process and revealing encrypted network traffic payloads. Figure 3 (next page) depicts the prototype network topology constructed for the purpose of resolving the research problem articulated for this thesis project.

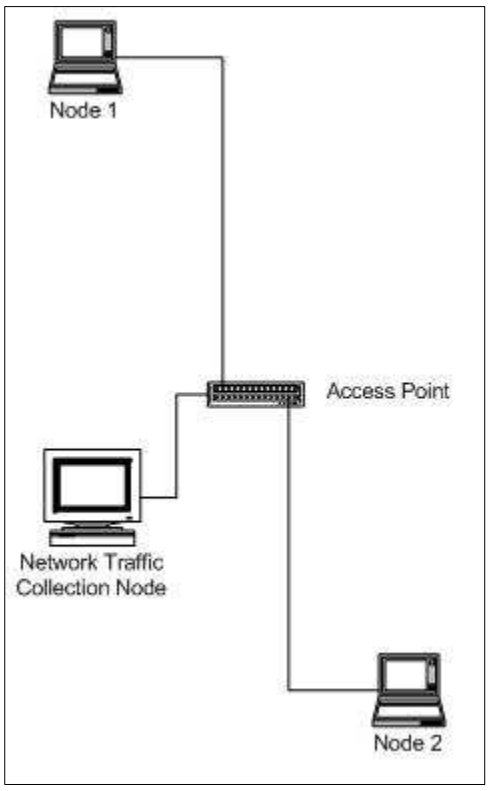


Figure 3: The logical layout of the prototype network constructed for resolving the research problem presented in this paper.

More than just integrating technology equipment for the expressed purpose of encrypting, transmitting, capturing, and deciphering network packets, the prototype network depicted in Figure 3 incorporates the majority of critical components considered necessary to relay electronic messages (encrypted or otherwise) between individual computers. At a minimum, computer networks – regardless of their intended purpose or physical footprint – require 1) individual nodes that communicate one with another, 2) network medium (either wired or wireless) upon which individual nodes place electronic messages to be transmitted to other nodes on the network, 3) dedicated switching equipment that facilitates efficient routing and transportation of electronic messages across network medium, 4) and communications protocols that govern how and when individual nodes package then transmit their respective electronic messages (Newton, 2006).

Because of its configuration – specifically with regards to the inclusion of A) encryption hardware and software and B) its general applicability to logical network topologies deployed and maintained by even the largest and most geographically diverse organizations – the network prototype architected for this project serves as a viable environment suited for answering the research question posed for this project. In answering the research question, network security professionals stand to enhance their ability to adequately resolve the encryption dichotomy, the value of which manifests itself as improved confidentiality and integrity of IT resources.

Procedure

As detailed in Chapter 3, successful analysis of encrypted network traffic ultimately requires knowledge of and access to the encryption key(s) originally employed in the process of converting clear-text messages into cypher text. The Access Point (AP) depicted in Figure 3 serves as the encryption key repository responsible for orchestrating the encryption of network traffic across the prototype topology that, subsequently, further facilitates eventual analysis of network traffic captured by the Network Traffic Collection Node (NTCN).

After configuring Node 1 (N1) and Node 2 (N2) with the same encryption key stored in the AP, N1 will transmit a malicious network message ("`I'm ready to install computer viruses!`" – no quotes) specifically addressed to N2 but over common communications medium shared by all devices on the network. Importantly, N1 will employ the encryption algorithm and encryption key delimited by the AP to transform the original, clear-text message into cypher text just prior to transmission; as such, the malicious message will traverse the network topology as packets containing nonsensical data to any device not sharing both the encryption algorithm and key.

Once the encrypted (yet still malicious) message arrives at N2, N2 will successfully reverse the encryption process using the encryption algorithm and key shared by N1 and the AP. Effective retransformation of N1's original message from cypher text back to clear text will automatically give N2 the advantage of clearly understanding the intent of N1's future actions.

Using software tool `tcpdump` previously installed on the NTCN, the NTCN will capture and locally store all host-to-host traffic traversing the network topology, including the encrypted network message originating from N1 and destined for N2. After capturing all network traffic, the NTCN will then rely on previously installed software tool `Wireshark` to graphically rebuild the network capture file locally stored to the NTCN. `Wireshark's` intuitive graphical user interface improves the efficiency at which network security professionals identify, dissect, and analyze host-to-host messages embedded in network capture files (Bejtlich, 2005).

Despite their collective value and potency as network monitoring and security tool, no amount of sophistication exempts `tcpdump` and `Wireshark` from the basic deciphering formula (decryption = algorithm + keys). While network security professionals may rely on `tcpdump` and `Wireshark` to trap, dissect, and analyze network messages, reconstructed encrypted network messages remain indiscernible cypher text up to the point where some procedure evokes the original encryption algorithm and key(s) to reverses the encryption process. Until such time as the applicable encryption key(s) decrypt their associated network messages, even powerful software tools like `tcpdump` and `Wireshark` cannot properly decode encrypted network packets into clear-text messages against which network security administrators may properly act.

Scenario 1

In the first procedural scenario devised for this project, the NTCN will capture and rebuild network traffic without the benefit of knowing the encryption algorithm and key configured into N1, N2, and the AP. Fundamental encryption principles suggest that without prior knowledge of the common encryption algorithm and key shared by N1, N2, and the AP, the NTCN (through the use of `Wireshark`) will unpack meaningless network messages that necessarily thwart reasonable courses of defensive action on the part of a network security professional charged with meaningful analysis of network traffic.

Scenario 2

Network security professionals intent on resolving the encryption dichotomy must maintain parity with network hosts receiving encrypted network messages. The attainment of such parity requires network security professionals – just like network hosts receiving encrypted network messages and successfully reversing the encryption process – to employ the same encryption key(s) stored in network equipment originally responsible for orchestrating encrypting network traffic.

In the second procedural scenario devised for this project, the NTCN will rebuild network traffic using `Wireshark` but, instead, will also administratively access the AP for the purpose of learning its configured encryption algorithm and retrieving its stored encryption key. Fundamental encryption principles suggest that correct identification of the AP's encryption algorithm and proper application of the AP's encryption key will result in the NTCN's ability to successfully decipher N1's malicious network messages.

Procedural Technique

Answering the research question devised for this thesis project depends on the degree to which commonly available technology components integrated into the prototype network topology support (or fail to support) host-to-host decryption processes. Since (as a mathematic rule) host-to-host decryption processes unequivocally rely upon proper application of applicable encryption keys, answering the research question devised for this thesis project necessarily rests on the retrieval and correct utilization of encryption keys stored in the AP.

Encryption Key Retrieval

The AP designed into the prototype network topology secures its stored encryption keys using authentication and authorization methods based on user name and password verification. Upon inputting correct login credentials, authorized collection of encryption keys merely requires secure navigation to the AP's embedded settings page responsible for delimiting the network encryption key.

Encryption Key Application

After acquiring the AP's network encryption key, the NTCN will load the encryption key into `Wireshark` but only after `Wireshark` rebuilds the network capture file originally created using `tcpdump`. Bejtlich (2005) and Orebaugh (2006) outlined basic implementations of `tcpdump` and `Wireshark` (respectively), although some configuration specifics relative to the prototype network designed for this research project necessitated modification to their rudimentary implementations.

Appendix C enumerates the configuration for each device incorporated into the prototype network topology constructed for this research project. In addition to itemizing hardware specification for N1, N2, the AP, and the NTCN, Appendix C also details the software tools

(including encryption algorithm and key) and associated command syntax used to generate, capture, and decode encrypted host-to-host network messages.

Data Analysis

Given that the ability for network surveillance and security specialists to successfully encrypt/decrypt network messages hinges on the acquisition and application of appropriate encryption algorithms and associated keys, the second procedural scenario devised for this project represents a plausible, viable step towards answering the research question and resolving the encryption dichotomy. Scenario 2 replicates an environment in which network security administrators may deploy host-to-host encryption schemes for the purpose of thwarting unauthorized eavesdropping but also confirms a procedure in which network security professionals maintain the ability to successfully reverse the host-to-host encryption process through measured retrieval of encryption keys.

Ability to A) retrieve encryption keys originally employed to created encrypted network packets and B) apply the same encryption keys to properly decrypt encrypted network packets captured in transit across a network topology serves as the primary method for answering the thesis question and will support the eventual formulation of a viable framework for network security professionals intent on resolving the encryption dichotomy. Actions A and B (above) represent quantifiable steps in the maturation of an encryption/decryption framework if only because inability on the part of legitimate, authorized network security administrators to successfully reverse the host-to-host encryption process necessarily prevents meaningful interpretation of network messages that, by unfortunate extension, obsoletes the need to move forward with finalizing a viable framework.

Chapter 5 – Results and Evaluation

Project Results – Scenario 1

Review of the network capture file generated by `tcpdump` using procedural governance parameters established for Scenario 1 revealed the incontrovertible accuracy of the basic deciphering formula (decryption = algorithm + keys). Figure 4 depicts the NTCN's `Wireshark` rebuild of the encrypted, malicious network message (see packet number 279) originating from N1 and destined for N2.

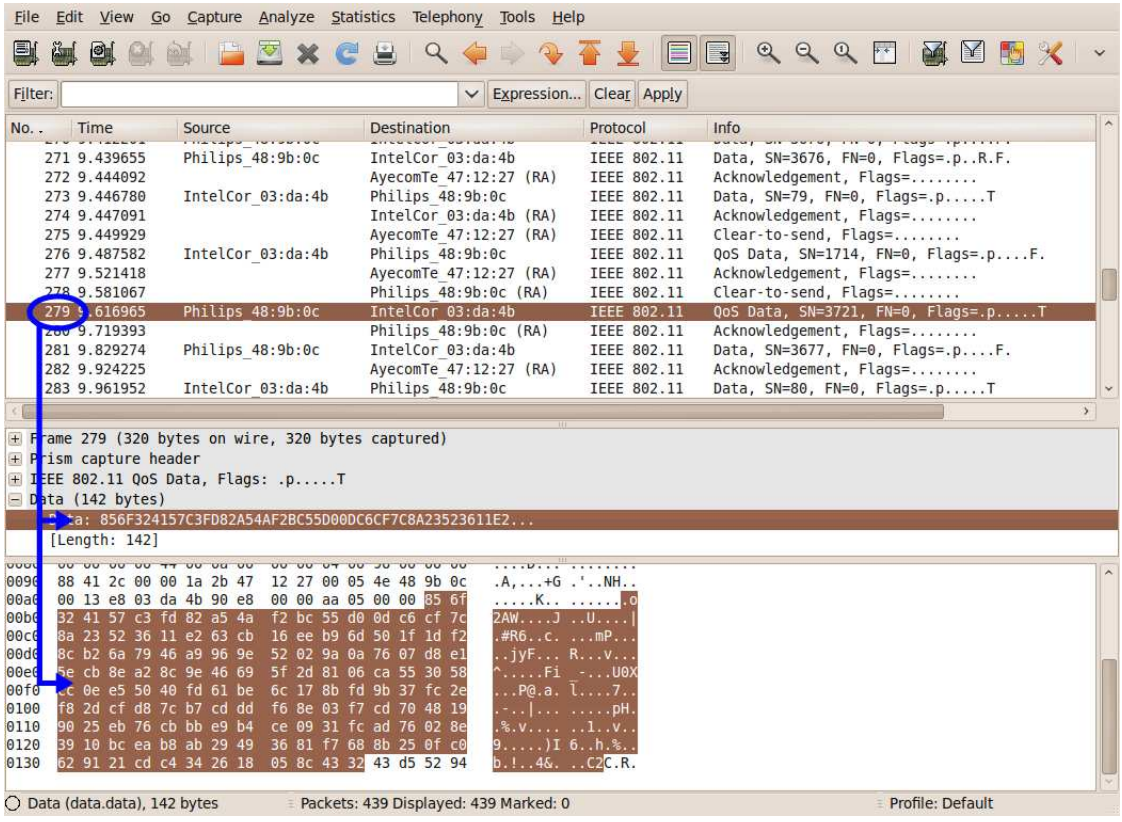


Figure 4: Node 1's encrypted network message as it appears to the Network Traffic Collection Node without application of the original encryption key.

The above figure illustrates the end result of N1 having transformed clear-text message `“I'm ready to install computer viruses!”` (no quotes) into the following cypher-text message prior to transmission across the prototype network topology:

```
856f324157c3fd82a54af2bc55d00dc6cf7c8a23523611e263cb16
eeb96d501f1df28cb26a7946a9969e52029a0a7607d8e15ecb8ea2
8c9e46695f2d8106ca553058cc0ee55040fd61be6c178bfd9b37fc
2ef82dcfd87cb7cdddf68e03f7cd7048199025eb76cbbbe9b4ce09
31fcad76028e3910bceab8ab29493681f7688b250fc0629121cdc4
342618058c4332
```

More importantly, without application of the original encryption key used by N1 to code its message destined for N2, The NTCN (using `Wireshark`) could *not* properly decipher N1's network message and reveal its malicious intent.

Project Results – Scenario 2

Review of the network capture file generated by `tcpdump` using procedural governance parameters established for Scenario 2 further reinforced the basic deciphering formula (decryption = algorithm + keys). Figure 5 (next page) depicts the NTCN's `Wireshark` rebuild of N1's encrypted, malicious network message, but Figure 5 also depicts correct deciphering of N1's encrypted, malicious network message (see packet number 279) through the NTCN's proper application of the correct encryption algorithm and key!

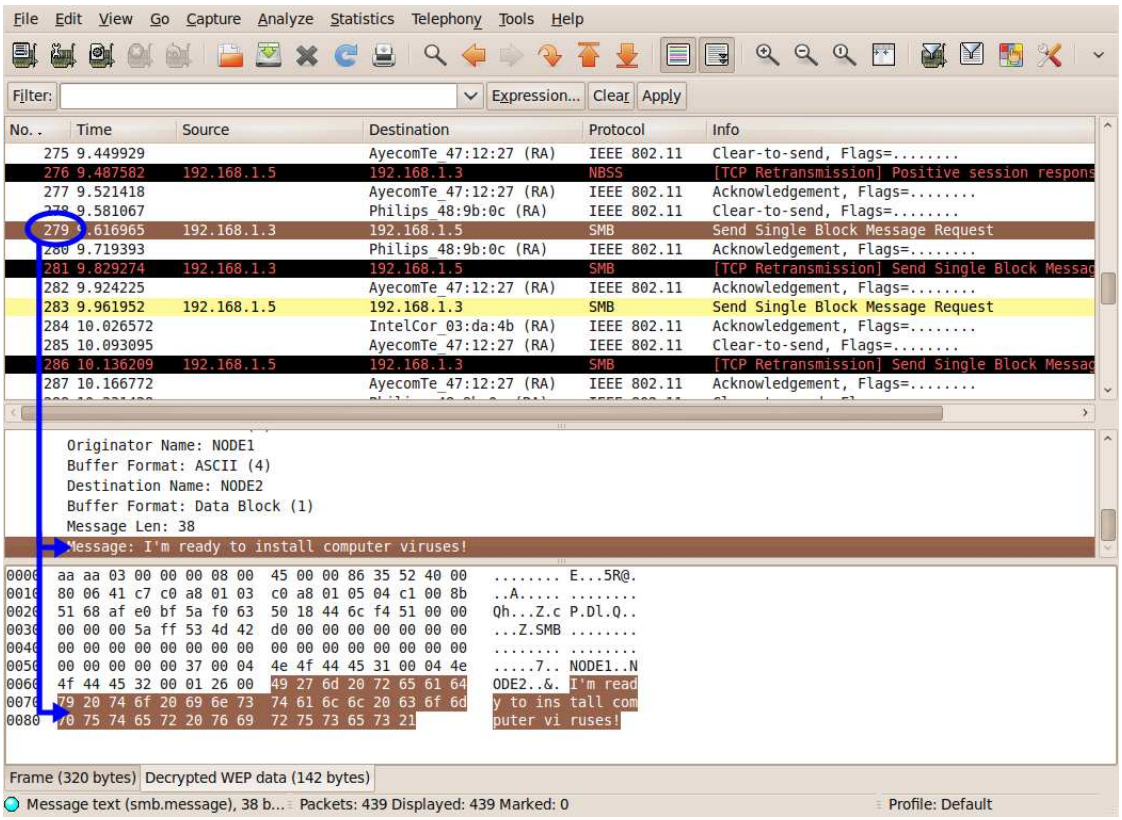


Figure 5: Node 1's encrypted network message as it appears to the Network Traffic Collection Node with application of the original encryption key.

Meaningful decoding of N1's encrypted network message required administrative access to the AP for the purpose of ascertaining its configured encryption algorithm and key that, by extension, dictated how all nodes associated with the network converted clear-text messages into cypher-text packets. With this critical information at its disposal, the NTCN meaningfully deciphered N1's malicious network message.

Results Analysis

Inability of the NTCN to reverse N1's encryption process (as exemplified by Scenario 1) resulted in indiscernible network messages. Encountered in production environments, such a scenario would severely frustrate attempts on the part of network security professionals trying to

uncover instances of malicious intent and execute subsequent courses of defensive action. Scenario 1 results exactly replicate the ineffective network surveillance environment outlined in Chapter 3 and serve to illustrate the serious risks associated with failure to resolve the encryption dichotomy.

Conversely, ability of the NTCN to successfully reverse N1's encryption process (as exemplified by Scenario 2) resulted in rapid discernment of N1's intent. Encountered in production environments, such a scenario would afford network security professionals the opportunity to execute courses of defensive action determined necessary to impede the installation of computer viruses (or any other type of malicious activity). Scenario 2 results exactly replicate the effective network surveillance environment outlined in Chapter 3 and, ultimately, demonstrate the benefits associated with resolving the encryption dichotomy.

Results Interpretation

Analysis of critical hardware and software encryption components commonly deployed across network topologies clearly demonstrates that proper access controls of encryption key repository devices within an administrative domain make significant contributions towards resolving the encryption dichotomy. When network security administrators maintain their ability to securely collect and properly apply encryption algorithms and keys deployed across network topologies, they can confidently employ host-to-host encryption schemes that insure only authorized decoding of in-transit network packets.

Results from this thesis project indicate that a forthcoming framework for maintaining defensibility in encrypted network environments must, at a minimum, include allowances for proper device access and control across discrete administrative domains. Because encryption

keys play such foundational roles in contemporary computer encryption/decryption algorithms, network security professionals intent on resolving the encryption dichotomy must continuously maintain administrative access to the device(s) – wherever their location and whatever their configuration – responsible for storing host-to-host encryption keys.

Authorized Administrative Access

Organizations committed to preserving electronic asset security through the inclusion of host-to-host encryption schemes must grant personnel responsible for conducting network surveillance (and subsequent emergency response) authorized administrative access to encryption key repository equipment. Depending on predetermined business rules adopted by individual organizations, collecting encryption keys could be a simple matter of job-duty assignment(s) or involve more complex invocations of an internal business process.

Even in the improbable (yet possible) event rogue hackers installed unauthorized key repository hardware or seized unauthorized control of legitimate key repository equipment on a given organization's network topology, the network engineering and administration team charged with maintaining defensibility have the option of removing (or rebuilding) the compromised devices simply because said devices fall within their jurisdictional domains. Decommissioning unauthorized or compromised encryption key repository equipment necessarily prevents hackers from using encryption keys external to administrative access of authorized network security team members and re-establishes administrative supremacy of authorized network administrators within their respective security domains.

Chapter 6 – Conclusion

Synopsis

The principle of visibility most applicably characterizes defensible, secure computer networks. In visible network environments, administrative and surveillance personnel maintain their continuous ability to **meaningfully inspect** individual message packets as they traverse network topologies. Importantly, security professionals rely on meaningful inspection of network messages to improve the efficiency at which they identify malicious activities and execute defensive courses of action commensurate with perceived threats.

Preserving visibility in unencrypted network environments proves fairly straightforward, but maintaining visibility in poorly conceived encrypted network architectures seriously undermines even the most thoughtfully executed defensive efforts of network security administrators. Encrypted network environments engineered and managed *without* consideration for encryption key retrieval methods desperately inhibit the abilities of authorized professionals to meaningfully inspect network packets for malicious activity thereby rendering their defensive capabilities thoroughly useless.

While network security engineers and administrators that design and defend encrypted network environments must plan for scenarios requiring proper decryption of coded host-to-host messages, no cohesive framework currently exists upon which IT professionals may rely that specifically addresses architecting such topologies. This thesis project contributes to a forthcoming framework by demonstrating: A) analysis of critical hardware and software encryption components commonly deployed across network topologies does, in fact, support

host-to-host decryption processes, B) allowances for authorized access to encryption key repository equipment substantially facilitates retrieval of encryption keys from critical hardware and software encryption components, and C) the ability to decipher previously encrypted network packets clearly improves the efficiency at which security professionals identify malicious activity.

Measured and meaningful reversal of the host-to-host encryption process and subsequent prescription of a critical framework parameter (i.e., authorized access to encryption key repository equipment) represent worthwhile steps towards developing a framework for maintaining defensibility in encrypted network environments. The successful formulation and execution of at least one prototype scenario that resolved the encryption dichotomy accentuates the prudence of moving forward to finalize a full-functioning encryption/decryption framework.

Futurecasting

Defensible problems manifested through the improper deployment of network encryption schemes only intensify as computer network technology matures and organizations evolve. This phenomenon has everything to do with the fact that the now-familiar Internet Protocol version 4 (IPv4) networking protocol – one of the core communications protocols that facilitates the overwhelming majority of today's Internet traffic – is quickly reaching the end of its available address space (Ford, 2010).

As an independent concept, diminished IPv4 address availability has very little impact on security administrators managing defensible network topologies. However, when combined with the fact that the next generation of Internet Protocol (IPv6) incorporates IPsec (a suite of encryption protocols) as an integral part of its host-to-host communication process, the inevitable

transition from IPv4 to IPv6 takes on monumentally greater significance. Whereas IPv4 affords the option to incorporate IPsec into network communication schemes, IPv6 mandates its inclusion (Frankel, 2010).

In such emerging environments, network security professionals that ignore resolution of the encryption dichotomy jeopardize their individual contributions towards maintaining defensible networks; moreover, failure to resolve the encryption dichotomy thwarts any given organization's ability to evolve and seriously compete on a global scale. In these regards, failure to resolve the encryption dichotomy 1) detrimentally impacts IT asset security and 2) hinders competitive advantage and economic growth potential.

Future Research

Devising a working scenario and prescribing a line-item framework inclusion that improves resolution of the encryption dichotomy represent only initial steps towards developing a comprehensive framework for maintaining defensibility in encrypted network environments. Potential topics for future research geared towards framework maturity include (but are not limited to):

1. Best-practice policies and procedures for encryption key creation and rotation.
2. Security strategies (both logical and physical) for preserving robustness and integrity of encryption key repository devices.
3. Internal business rules (likely based on the principle of separation of duties) dictating whom may access encryption key repository equipment.
4. Internal business rules for best delimiting what events necessitate authorized access of encryption keys to reverse the host-to-host encryption process.

5. Legal ramifications possibly resulting from eavesdropping – even authorized eavesdropping – on encrypted network traffic.
6. Practicality of encryption key retrieval methods to all levels of the OSI network model.
7. Inclusion of asymmetric encryption methodologies.
8. Applicability to virtual computing environments.

Development of a full-functioning encryption/decryption will require contributions from private business, government agencies, and academic institutions. Such an approach will insure the greatest applicability to unique needs associated with diverse IT industry segments.

Conclusion

Network defensibility without consideration for network visibility results in wholly ineffectual network security. Otherwise defensible network architectures that fail to resolve the encryption dichotomy axiomatically increase IT asset vulnerability if only because security professionals lose effective visibility of device communications.

The ability to meaningfully inspect <u>every</u> network packet that traverses a given IT topology constitutes a key characteristic of visible network architectures. Network security engineers and administrators that design and maintain encrypted network environments must plan for inevitable instances where IT asset security hinges on the ability to successfully decrypt previously encrypted host-to-host network messages, meaningfully ascertain malicious intent, and launch effective courses of defensive action.

References

Ambler, S. W. (n.d.). *Agile modeling: a brief overview*. Web. 2, March, 2011. <http://subs.emis.de/LNI/Proceedings/Proceedings07/AgilModel_aBrief_1.pdf>.

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the Association for Information Systems*, v2 a19, 1 – 32.

Bejtlich, R. (2005). *The tao of network security monitoring*. Boston: Addison-Wesley.

Bernard, S. A. (2005). *An introduction to enterprise architecture* (2nd ed.). Bloomington, IN: AuthorHouse.

Burnett, S. & Paine, S. (2004). *RSA Security's official guide to cryptography*. New York: McGraw-Hill.

Ciampia, M. (2009). *Security+ guide to network security fundamentals*. Boston: Course Technology.

Cockburn, A. (2003). *Research methods in information systems research: matching method to researcher* [Electronic version]. Web. 24, January, 2011. <http://alistair.cockburn.us/get/2423>.

Elucidata Services, (n.d.). *The software development life cycle (SDLC) for small to medium database applications*. Web. 2, March, 2011. <http://www.elucidata.com/refs/sdlc.pdf>.

Ford, M. *Are you ready for the big Internet crunch?* Cable News Network, May 28, 2010. Web. 17, June, 2010. <http://www.cnn.com/2010/TECH/05/27/internet.crunch.2012/index.html>.

Forouzan, B. A. (2008). *Cryptography and network security*. Boston: McGraw-Hill.

Forward, K. (2002). *Appropriate use of network encryption technologies* [Electronic version]. Bethesda, MD: The SANS Institute.

Frankel, S., Graveman, R. & Pearce, J. (2010). *Guidelines for the secure deployment of IPv6 (draft)*. Gaithersburg, MD: National Institute of Standards and Technology.

Frankel, S., Kent, K., Lewkowski, R., Orebaugh, A. D., Ritchey, R. W. & Sharma, S. R. (2005). *Guide to IPsec in VPNs*. Gaithersburg, MD: National Institute of Standards and Technology.

Garman, J. (2003). *Kerberos: the definitive guide*. Beijing: O'Reilly & Associates, Inc.

Gebski, M., Penev, A. & Wong, R. A. (2006). *Protocol identification of encrypted network traffic*. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. Web. 1, September, 2010 <http://portal.acm.org/citation.cfm?id=1249160>.

Koch, R. & Rodosek, G. D. (2010). Command evaluation in encrypted remote sessions. *Network and Systems Security*. September 2010, 299 – 305.

Leedy, P. D. & Ormrod, J. E. (2005). *Practical research planning and design*. Upper Saddle River, NJ: Pearson Prentice Hall.

Lewand, R. W. (2000). *Cryptological mathematics*. Washington, D.C.: The Mathematical Association of America.

Liu, P., Zong, R. & Liu, S. (2008). A new model for authentication and authorization across heterogeneous trust-domain [Electronic version]. *2008 International Conference on Computer Science and Software Engineering*, December 12 – 14, 2008, 789 – 792.

Mackey, D. (2003). *Web security for network and system administrators*. Boston: Course Technology.

Mel, H. X. & Baker, D. (2001). *Cryptography decrypted*. Boston: Addison-Wesley.

Newton, H. (2006). *Newton's telecom dictionary (22nd ed.)*. New York: CMP Books.

Orebaugh, A., Ramirez, G. & Beale, J. (2006). *Wireshark and Ethereal network protocol analyzer toolkit*. Burlington, MA: Syngress.

Oxenhandler, D. (2003). *Designing a secure local area network* [Electronic version]. Bethesda, MD: The SANS Institute.

PCI Security Standards Council, LLC. (2009). *Payment Card Industry (PCI) Encrypting PIN PAD (EPP) security requirements (version 2.1)*. Wakefield, MA: Author.

Pfleeger, C. P. & Pfleeger, S. L. (2007). *Security in computing*. Boston: Pearson Education, Inc.

Ross, J. (2004, October). Generating strategic benefits from enterprise architecture. *Research Briefing*, v4 n3A, Cambridge, MA: MIT Sloan School of Management.

Ross, J. (2005, July). Understanding the benefits of enterprise architecture. *Research Briefing*, v5 n2B, Cambridge, MA: MIT Sloan School of Management.

Schneier, B. (1996). *Applied cryptography*. New York: John Wiley & Sons, Inc.

Styer, E. (n.d.). *JavaScript DES example*. Web. 5, March, 2011. <http://people.eku.edu/styere/Encrypt/JS-DES.html>.

Wright, C. V., Monrose, F. & Masson, G. M. (2006). On inferring application protocol behaviors in encrypted network traffic. *Journal of Machine Learning Research*, v7, 2745 – 2769.

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal*, v38 n2&3, 454 – 470.

Appendix A

Converting Clear Text Into Cypher Text

Although the US federal government first adopted the Data Encryption Standard (DES) in 1974 (Mel, 2001), individual consumers, private retailers, and financial institutions of all shapes and sizes continue to rely on DES and its matured variants (like 3DES) to protect banking transactions totaling trillions of dollars (PCI, 2009). Similarly, common network security protocols and architectures (e.g., IPsec and Kerberos) rely on DES variants to defend against effective eavesdropping of network traffic payloads (Frankel, 2005 & Garman, 2003, respectively).

Unfortunately, just as legitimate network users and administrators rely on DES to protect authentic communications, so too can malicious attackers use DES to hide and protect their criminal activities. The following steps explain the process evoked to obfuscate the malicious, clear-text message introduced in Table 1 into cypher text using the DES algorithm.

Step 1:

In order to encrypt `"I'm ready to install computer viruses!"` (no quotes) using the web-based DES encryption/decryption tool offered by Eugene Styer (n.d.), the original message had to be segmented into packets 64 bits (or 8 ASCII characters) in length (no more, no less). Spaces (delimited as `<space>` in Table 6) represent one and only one ASCII character value and were sometimes used to pad packets in order to satisfy strict packet length requirements of Styer's web-based tool. In the interest of future replication, the web-based tool was configured as follows:

1. Input message = ASCII
2. DES encryption key = `3b3898371520f75e` (hexadecimal value)
3. DES encryption key = `00111011 00111000 10011000 00110111 00010101 00100000 11110111 01011110` (*non-configurable* binary value)
4. Output message = Hexadecimal

Packet Order	**Input Packet Payload** (8 ASCII characters)	**Resulting Binary Value** (non-encrypted)
First Packet	I'm<space>read	01001001 00100111 01101101 00100000 01110010 01100101 01100001 01100100
Second Packet	y<space>to<space>ins	01111001 00100000 01110100 01101111 00100000 01101001 01101110 01110011
Third Packet	tall<space>com	01110100 01100001 01101100 01101100 00100000 01100011 01101111 01101101
Fourth Packet	puter<space>vi	01110000 01110101 01110100 01100101 01110010 00100000 01110110 01101001
Fifth Packet	ruses!<space><space>	01110010 01110101 01110011 01100101 01110011 00100001 00100000 00100000

Table 6: The malicious network message introduced in Table 1 properly segmented then correctly converted to binary values

IMPORTANT: The resulting binary values delimited in Table 6, column 3 do *not* constitute encrypted messages. Step 1 merely represents the conversion of alpha-numeric characters (ASCII code) best recognized by humans into binary values (1s and 0s) best recognized by computers. Because XOR-ing operations technically work against individual computer bits (i.e., 1s and 0s) in network packets, converting ASCII characters to binary values improves clarification of the XOR process explained in Appendix B.

Step 2:

With message packets correctly segmented to conform to strict tool requirements, the web-based tool then ran each packet through a DES encryption engine using the DES encryption

key notated above (`3b3898371520f75e`). Because of pre-selected configuration parameters (see above), the web-based tool output each encrypted packet in hexadecimal notation.

Packet Order	**Packet Payload** (Binary values from Step 1)	**Operation**	**DES Encryption Key**	**Resulting Hexadecimal Value** (encrypted)
First Packet	01001001 00100111 01101101 00100000 01110010 01100101 01100001 01100100	XOR	3b3898371520f75e	bb9ca9aa479de85c
Second Packet	01111001 00100000 01110100 01101111 00100000 01101001 01101110 01110011	XOR	3b3898371520f75e	b80397ebe29742f6
Third Packet	01110100 01100001 01101100 01101100 00100000 01100011 01101111 01101101	XOR	3b3898371520f75e	7163182e01941f1c
Fourth Packet	01110000 01110101 01110100 01100101 01110010 00100000 01110110 01101001	XOR	3b3898371520f75e	05b59a4469632c6e
Fifth Packet	01110010 01110101 01110011 01100101 01110011 00100001 00100000 00100000	XOR	3b3898371520f75e	cc869012ba3d0462

Table 7: The malicious network message introduced in Table 1 XOR-ed against the DES encryption key introduced in Table 2. The resulting message is fully encrypted.

IMPORTANT: The resulting hexadecimal values delimited in Table 7, column 5 constitute encrypted messages, because they were XOR-ed against an encryption key using the Feistel function that serves as the primary XOR engine of the DES algorithm (Forouzan, 2008).

Step 3:

Finally, all encrypted hexadecimal values from each packet (as notated in column 5 of Table 7) were then combined into one hexadecimal string such that the clear-text message `"I'm ready to install computer viruses!"` (no quotes) became the encrypted message: `bb9ca9aa479de85c b80397ebe29742f6 7163182e01941f1c 05b59a4469632c6e cc869012ba3d0462`.

Tables 4 & 5:

Due to configuration restrictions embedded in the web-based tool utilized to complete steps 1 through 3 in this appendices, the DES encryption key originally used to scramble the clear-text message could not be altered. Consequently, a different encryption key using a simple XOR function (in the case of Table 4) and the same encryption key using a simple AND function (in the case of Table 5) were employed to illustrate how variations to either the encryption key or the mathematic operation necessarily result in indiscernible, worthless messages during the decryption process.

Appendix B

XOR Operations

Computer scientists and computer programmers rely on Exclusive OR (XOR) operations to determine the disjunction between two operands. In the case of the malicious network message introduced in Table 1, one operand equals the clear-text message (`I'm ready to install computer viruses!`) and the second operand equals the DES encryption key (`3b3898371520f75e`). Although the following explanation oversimplifies the XOR engine embedded in DES, it accurately demonstrates (without too much imagination) how XOR-ing a clear-text message against an encryption key results in cypher text.

Technically, XOR operations compare bit values of two different operands using the following truth table:

First bit value	Operand	Second bit value	Truth Result
1	XOR	1	0
1	XOR	0	1
0	XOR	1	1
0	XOR	0	0

Table 8: Exclusive OR (XOR) truth table

Step 1:

Start with a simple clear-text ASCII message. To simplify this explanation, the word "computer" (no quotes) serves as the clear text message (or first operand).

```
computer
```

Step 2:

Convert the clear text message into its binary equivalent. This step facilitates bit comparison as required by the above XOR truth table.

```
Computer = 01100011 01101111 01101101 01110000 01110101 01110100
01100101 01110010
```

Step 3:

Convert the encryption key (or second operand) into its binary equivalent. As with Step 2, this step facilitates bit comparison as required by the above XOR truth table:

```
3b3898371520f75e = 00111011 00111000 10011000 00110111 00010101
00100000 11110111 01011110
```

Step 4:

Use the above XOR truth table to arrive at the binary results of having operated the clear-text message against the encryption key.

```
Text:  01100011 01101111 01101101 01110000 01110101 01110100 01100101 01110010
Key:   00111011 00111000 10011000 00110111 00010101 00100000 11110111 01011110
Result:01011000 01010111 11110101 01000111 01100000 01010100 10010010 00101100
```

Step 5:

Convert the binary result from Step 4 (highlighted above) back into ASCII clear text.

```
XwõG`T□,
```

At this point, the original clear-text message as been “encrypted” into a meaningless message. Only a person (or computer) with access to the original, unadulterated encryption key could easily reverse the steps detailed above to revert back to the original, intelligible clear-text message.

Appendix C

Prototype Network Specifications

Tables 9 through 11 (below) detail the hardware and software specifications for each device incorporated into the prototype network topology. Importantly, Node 1, Node 2, and the Network Traffic Collection Node include hardware (e.g., IBM and Intel) and software (e.g., Microsoft Windows and Ubuntu Linux) IT solutions common to wide varieties of private corporations, public organizations, and government agencies.

Nodes 1 & 2:

	Node 1 System Summary	**Node 2 System Summary**
Manufacturer	IBM	IBM
Model #	T42 Type 2373	T42 Type 2373
System serial #	99FR34Z	L388G13
System board serial #	J1X6N4AF1VJ	VJ0BU5C619R
BIOS version	3.23 (1RETDRWW)	3.23 (1RETDRWW)
BIOS date	2007-06-18	2007-06-18
OS	Microsoft Windows XP	Microsoft Windows XP
OS version	5.10.2600 Service Pack 3	5.10.2600 Service Pack 3
Processor	x86 Family 6 Model 13 Stepping 6	x86 Family 6 Model 13 Stepping 6
Main memory	2096 mb	2096 mb
Drive type & size	Fixed; 80 gb	Fixed; 80 gb
Network device	Atheros 11a/b/g Wireless LAN Mini PCI Adapter	Intel PRO/1000 MT Mobile Connection
Network driver version	4.1.2.156	8.10.3.0
Network hardware address	00:05:4e:48:9b:0c	00:13:e8:03:da:4b
Network logical address (DHCP)	192.168.1.3	192.168.1.5

Computer network name	Node1	Node2
Configuration Notes: 1. All Microsoft Windows “High Priority” and “Hardware” updates confirmed current via Windows Update manager (as of March 19, 2011). 2. No reported errors in Device Manager. 3. Embedded firewall disabled; no anti-virus software installed. 4. Microsoft's “Messenger Service” enabled to facilitate host-to-host network communication. 5. DOS network message command syntax (from Node 1 to Node 2) : `net send NODE2 “I'm ready to install computer viruses!”`		

Table 9: Detailed specifications for Node 1 and Node 2 incorporated into the prototype network topology.

Access Point:

	Access Point (AP) System Summary
Manufacturer	Actiontec
Model #	Q1000
System serial #	CVAA9202505823
Firmware version #	QA02-31.10L.48
Network hardware address	00:24:7b:e2:6b:e0
Network logical address	192.168.1.1
DHCP server	Enabled
SSID	Regis
Encryption protocol	Wired Equivalent Privacy (WEP)
Encryption key	3b38983715

Table 10: Detailed specifications for the Access Point incorporated into the prototype network topology .

Network Traffic Collection Node:

	Network Traffic Collection Node (NTCN) System Summary
Manufacturer	Asus
Model #	EEEPC901-BK001
System serial #	87OAAQ354129
BIOS version	2103
BIOS date	06/11/09

OS	Ubuntu Linux 9.04; GNOME 2.26.1
Kernel version	2.6.29-1-netbook
Processor	Intel Atom CPU N270 x 2
Main memory	2006 mb
Swap space	1309 mb
Drive type & size	Fixed; 32 gb
Network device	RaLink RT2860
Network hardware address	00:15:af:ca:fd:1e
Network logical address (DHCP)	Set to *monitor* mode; no IP address assigned
Configuration Notes: 1. Update Manager confirmed all installed packages current (as of March 19, 2011). 2. Tcpdump 3.9.8-4ubunut installed. 3. Wireshark 1.0.7 installed. 4. UNIX command syntax to capture network packets: `tcpdump -n -i ra0 -s 0 -w <capture.file.name>` 5. Wireshark application of network encryption key: Step A: Open > capture.file.name Step B: Edit > Preferences > Protocols > IEEE 802.11 > insert.encryption.key	

Table 11: Detailed specification for the Network Traffic Collection Node incorporated into the prototype network topology.