DEFEATING ADVERSARY NETWORK INTELLIGENCE EFFORTS WITH ACTIVE CYBER DEFENSE TECHNIQUES

GRADUATE RESEARCH PAPER

Keith A. Repik, Major, USAF

AFIT/ICW/ENG/08-11

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this graduate research project are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

AFIT/ICW/ENG/08-11

DEFEATING ADVERSARY NETWORK INTELLIGENCE EFFORTS WITH ACTIVE CYBER DEFENSE TECHNIQUES

GRADUATE RESEARCH PAPER

Presented to the Faculty

Department of Electrical & Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Cyber Warfare

Keith A. Repik, BS, MS, MA

Major, USAF

June 2008

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

AFIT/ICW/ENG/08-11

DEFEATING ADVERSARY NETWORK INTELLIGENCE EFFORTS WITH ACTIVE CYBER DEFENSE TECHNIQUES

Keith A. Repik, BS, MS, MA

Major, USAF

Approved:

6 Jun 08

Maj Paul D. Williams, PhD, USAF (Chairman) Date

6 JUN 08

Dr. Robert F. Mills (Member) Date

AFIT/ICW/ENG/08-11

Abstract

The purpose of this research was to assess the potential of dynamic network reconfiguration and decoys to defeat adversary network reconnaissance efforts, thereby improving network defense. Specifically, this study sought to determine if the strategy has merit, thus warranting more resource intensive research and engineering studies. The research objective was achieved through a comprehensive literature review and technology survey. The key topics examined in the literature review include the network attack process, network defense strategies, deception, obfuscation, and the concept of continuous unpredictable change. Many candidate technologies were surveyed, but only three, identified as high potential, were examined in detail: address hopping, honeypots and network telescopes.

The following conclusions were reached: (a) the concept has merit and should be pursued further – dynamic network reconfiguration and decoys have demonstrated effectiveness in controlled experiments; (b) it's achievable in the near term – the essential technologies are available today; and (c) extensive analysis and engineering is needed to determine which technologies are appropriate, how and where to integrate them into our networks and how to employ them most effectively.

Table of Contents

Abstract v

Table of Contents vi

List of Figures viii

List of Tables ix

I. Introduction 10

Background 10
Motivation 10
Purpose 12
Scope 12
Results 12
Thesis Organization 13

II. Defeating the Network Attack Process 14

General Network Attack Methodology 14
Attacking the Methodology 16
Obfuscation 17
Deception 18
Overview 18
Fundamentals 19
Considerations 20
Deception in Network Defense 21
Dynamic Network Reconfiguration 22

III. Technologies of Interest 24

Address Hopping 24
Overview 24
Effectiveness 26
Technical Overview 32
Integration with Encryption Services 39
Performance and Interoperability Considerations 40
Defeating Address Hopping 42
Honeypots 43
Overview 43
Detection 44
Fake Honeypots 46
Network Telescopes 46

IV. Conclusion ……. 48

Findings 48
Future Research 49

Bibliography ……. 52

Vita ……... 56

List of Figures

Figure Page

Figure 1: General Attack Methodology 15

Figure 2: DARPA Experimental Results 28

Figure 3: LAN Segment with Distant Server 34

Figure 4: LAN Segment with Distant Server 34

Figure 5: LAN Segment with Local Server 35

Figure 6: Local Router to Local Router 36

Figure 7: Gateway to Gateway 37

Figure 8: Address Hopping Mechanism Placement 37

Figure 9: LAN Segment to LAN Segment 38

List of Tables

Table Page

Table 1: Sample Address Hopping Use Case.....33

AFIT/ICW/ENG/08-11

DEFEATING ADVERSARY NETWORK INTELLIGENCE EFFORTS
WITH ACTIVE CYBER DEFENSE TECHNIQUES

I. Introduction

Background

The majority of network defense activity is currently focused on defeating attacks and recovering from their effects. However, the number of known vulnerabilities continues to rise with increasing system complexity, while exploit development time remains far shorter than patch development and deployment time, leaving systems increasingly exposed. Acknowledged and suspected Department of Defense information system intrusions – Moonlight Maze, Titan Rain, etc – continue to increase while trust in our own systems declines. Something more is needed.

Motivation

This research supports an Air Force Cyber Command (AFCYBER (P)) requirement. A series of internal discussions were held by AFCYBER (P) staff on the nature of emerging threats, changing technology, and the limitations of current network defense techniques. They concluded that the static nature of our networks presents a 'sitting target' – ceding the initiative and limiting our ability to maneuver. Adversaries are able to build detailed target folders, and attack with precision at the time and place of their choosing. It was theorized that adding mobility and uncertainty would greatly enhance our network defenses, degrading (or defeating) the adversary's Intelligence Preparation

of the Network Battlespace (IPB). Dynamically changing network configurations and employment of decoys were recommended to achieve the desired effect.

AFIT research support was requested to further develop the concept, assess its merit, and explore potential technology solutions. Initial AFIT analysis yielded the following proposals:

- Dynamically create decoys and sensors in the dark (unused) regions of our Internet Protocol (IP) address space.
- Move our networks around in IP address space – either continuously or when directed (ex: 'war reserve' mode).
- Dynamically and continually move our network elements – servers, hosts, etc – around the entire IP address range.
- Combine all three use cases for greatest effect.

This proposal is well illustrated using the defensive tactics of mobile ballistic missile launchers (SCUDs) as a rough analog. An effective tactic used to avoid detection and destruction is periodic, unpredictable relocation of the launch sites [20: 32]. The ultimate objective is to preserve friendly combat power, while eroding or wasting the adversary's combat power. To succeed defenders must be able to continue effective operations, while attackers are unable to perform reconnaissance and effectively strike within a single relocation cycle. Given sufficient resources the adversary will locate and destroy the sites. However, those resources can no longer be employed elsewhere, thus the price of success may be higher than the adversary is willing or able to pay.

Supplementing this tactic with decoy launch sites, both fixed and mobile, increases its effectiveness by forcing the adversary to examine a larger target set – potentially striking

incorrect targets or striking ineffectively [20: 32]. Examination of all candidates, even if perfunctory, is required to separate real targets from decoys. Success or failure of the deception merely changes the amount of resources wasted by the adversary. Finally, sensors – both fixed and mobile – provide friendly forces critical feedback on adversary activities and insight into their tactics, techniques and procedures.

Purpose

The purpose of this research project is to assess the potential effectiveness of dynamic network reconfiguration and decoys for network defense. Specifically, their ability to hinder the adversary's network reconnaissance process will be examined.

Scope

This is an initial assessment only – a review of existing research, overview of underlying theories, examination of several key technologies and a determination of merit. A full assessment will require additional research, in-depth examination of many technologies, detailed engineering studies and cost-benefit analyses. These are beyond the scope of this effort.

Results

The following conclusions were reached: (a) the AFCYBER (P) concept has merit and should be pursued further; (b) it's achievable in the near term – the essential technologies are available today; and (c) extensive analysis and engineering is needed to determine which technologies are appropriate, how and where to integrate them into our networks and how to employ them most effectively.

Thesis Organization

This chapter presents the motivation, purpose, scope and results for this research, and concludes with the document's organization. Chapter 2 begins exploring the basis of the findings, providing an overview of the general network attack methodology, adversary exposure points in the process, and the effectiveness of obfuscation and deception – especially for disruption of network reconnaissance. Chapter 3 discusses several key technologies supporting the AFCYBER (P) concept – address hopping, honeypots, and network telescopes. Chapter 4 contains the conclusion and ideas for future research.

II. Defeating the Network Attack Process

General Network Attack Methodology

Well-resourced, risk-averse, skilled adversaries – such as nation-states and some state-sponsored organizations – typically attack networks in a stealthy manner to either capture sensitive information or disrupt normal operations. These actions are tied to immediate operations and/or preparation for future conflicts. Unlike amateurs, they are disciplined and generally do not attack for sport. As professionals, they conduct reconnaissance, plan their operations to achieve specific goals, and prioritize their activities in accordance with time, budget and other resource constraints [6: 1].

The general approach used by professionals to attack networks is well-documented and understood – intelligence gathering and target identification, initial planning and development, network reconnaissance, follow-on planning and preparation, attack and damage assessment. This process has decision points controlling when to proceed, loop back or terminate operations (Figure 1). Successfully influencing adversary decision making at these keys points could result in action favorable to the defender [9: 28; 6: 1].

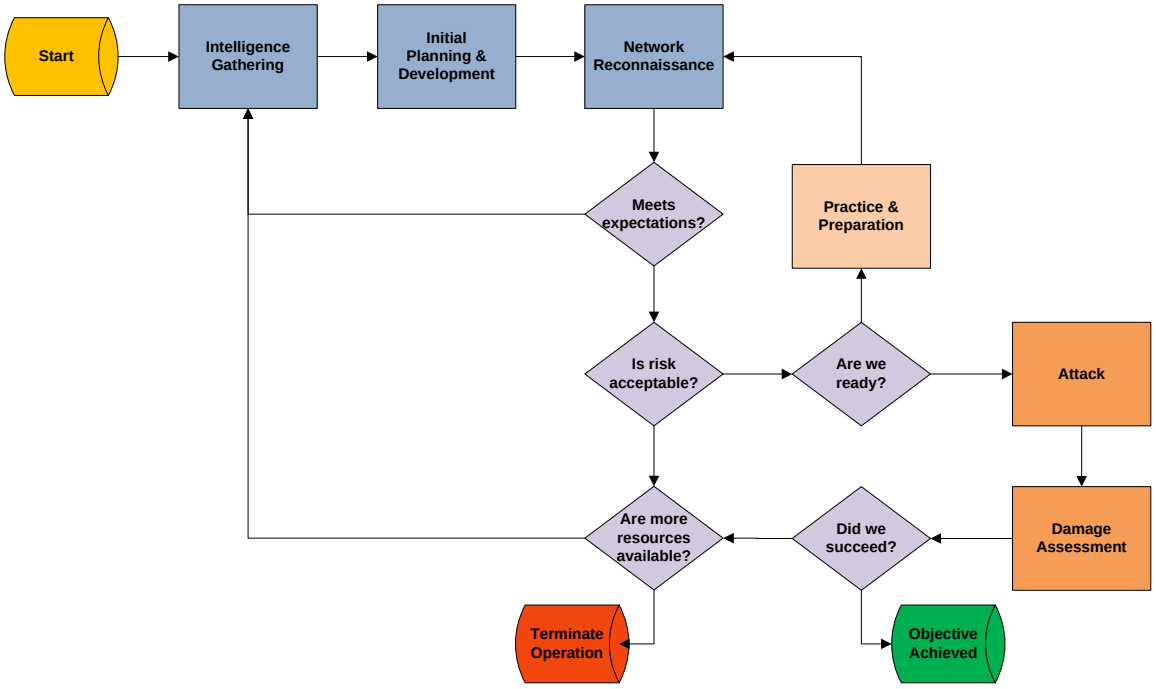

Figure 1: General Attack Methodology [6]

The first step is identification and selection of a target. Adversaries gather intelligence on candidates through direct observation, investigation, and by learning from other people or agents [19: 26]. Analysis of this information enables adversaries to conduct an initial cost-benefit analysis, weigh potential gains and risks, and select a target. Following target selection, initial planning and preparation occurs to minimize risk. Any preliminary information obtained on the target – such as organizational mission, staffing and key personnel, technical sophistication, partner organizations, web presence, etc – may shed insight into the network type, its weaknesses and the defensive tactics, techniques and procedures employed. This enables mission planners to allocate appropriate resources and operators to select appropriate tools for the next phase – network reconnaissance.

Conducted to gather detailed technical information on the target and visualize it, network reconnaissance is often the most time-consuming [6: 2] due to the desire for stealth and certainty. The critical first step is identification of live hosts. Once located, identification of associated applications, services, and operating system type and version can be accomplished. Finally, the information gathered is used to visualize the network architecture – network segments, routers, firewalls, production systems, etc. Powerful, actively evolving tools – such as NMAP and Nessus – are freely available to support network reconnaissance [9: 28].

Intelligence gained during network reconnaissance supports critical decisions on whether to advance, continue preparing or withdrawal. If the decision to proceed is made, the attack and damage assessment phases are entered. Network attacks can consist of a wide variety of acts depending on the objective. Examples include denial of service, malware installation, and information exfiltration, corruption or destruction. Upon completion, the adversary assesses the results to determine if additional action is required or the objective has been achieved.

Attacking the Methodology

Like terrorist actors, network attackers are most vulnerable at two points in the attack process – reconnaissance and attack. As previously noted, most network defense activity is focused on defeating and recovering from the attack, with comparatively little investment in defeating reconnaissance efforts. However, during reconnaissance, adversaries are exposed to risk for a longer period of time and are often less confident due to lack of information. As such, they are more susceptible to deterrence [4: 10]. By some estimates, 95% of an adversary's time is spent preparing for an attack, while only

5% is spent actually executing it [6: 2]. Defeating or degrading network reconnaissance efforts – and thus preventing or weakening subsequent attacks – appears to be a low risk and potentially highly effective defensive tactic.

Most network reconnaissance defense techniques attempt to hide the network behind screens – such as firewalls and proxy servers. The general strategy is that if the adversary can't 'see' the network, they can't gather meaningful intelligence on it. However, these defenses are most effective against external adversaries only, while proximate adversaries – trusted or untrusted insiders – are generally unaffected. Though other countermeasures can be deployed to cover this gap, evidence to date suggests that skilled persistent adversaries will obtain any information given enough time. An alternative strategy is therefore to limit the useful life expectancy of the information through continuous unpredictable changes [3: 9]. Often considered a form of obfuscation, it can be implemented through dynamic network reconfiguration techniques and is particularly effective when employed in concert with deception measures such as decoys.

Obfuscation

The purpose of obfuscation is to confuse an adversary, increasing the effort required to discern protected content [21]. Many methods to obfuscate content exist, to include format alternation, scrambling identifiers, and reordering content. Among its many uses, it's commonly used to protect software and communications. For example, malware developers use it to complicate reverse engineering – rendering software unintelligible but still functional [22]. Steganography – hiding secret content within innocuous open content – is used to create watermarks and hide messages within pictures. While not as

robust as strong cryptography, obfuscation can greatly increase the effort required to locate and correctly interpret protected content.

Deception

Overview

Deception has historically been a vital component of warfare. Written around 800 B.C, the first known treatise espousing the art of deception in warfare is Sun Tzu's "The Art of War" [4: 3]. It has proven useful for four general reasons. First, it increases friendly freedom of action by drawing the adversary's focus of attention away from the real action being taken elsewhere. Second, it may cause the adversary to adopt a disadvantageous course of action. Third, it can help gain the element of surprise. Fourth, deception can preserve friendly resources [14: 1] while confusing or overloading adversary intelligence capabilities [4: 11].

The effectiveness of deception has persisted, despite the evolution of warfare, because it's fundamentally about human nature and perception, rather than technology. Warfare is a human activity, directed by human decision-makers. Researchers have repeatedly demonstrated that people are, in general, poor at detecting deception [16]. Of significance to both network attackers and defenders, they're even worse at detecting online deceptions [5] – as witnessed be the effectiveness of 'social engineering' and 'spear phishing' computer attacks; and, evidence that many traditional scams are more effective on the Internet than in the real world [12: 29].

While highly effective if successful, deception is not without drawbacks. It can be resource intensive and risky. Skilled personnel, detailed planning and extensive intelligence support are required. If the deception fails, these resources are wasted.

Unfortunately, deception may fail for a variety of reason, many not controllable – the target may not perceive the deceit, be unable to act, be indecisive, act in unforeseen ways, or discover the deception. When planning operations, the effects expected with success must therefore be carefully weighed against the resources required and risks associated with failure [8].

Fundamentals

All deceptions stem from the ability to influence adversary observables and rely on an assumption of finite resources. Deception is based on showing and hiding these observables [19: 28]. Observables represent what the adversary can sense through their Intelligence, Surveillance and Reconnaissance (ISR) assets. With finite resources – ISR assets, intelligence analysts, time, money, etc – adversaries cannot be in all places at all times and are vulnerable to resource exhaustion. Deception thus consists of (a) determining what the adversary should and shouldn't observe, (b) creating simulations to induce desired observations and control the focus of attention, and (c) using concealment to inhibit undesired observations [4: 14].

Effective deceptions exploit preexisting beliefs of the adversary, with expectations playing a key role in their susceptibility. Large deviations from expected patterns will draw attention and may trigger suspicion. Deceptions which match familiar patterns will likely result in the adversary following the expectations of that pattern. If the objective is to draw attention to the deception, then large deviation is desirable. If the objective is to avoid notice or evoke a predicable response, less observable deviation is more likely to succeed [4: 15]. Inadequate understanding of adversary expectations is a prime reason for deception failure.

Per Joint Publication 3-13.4, Military Deception (2006), the ultimate objective of deception is to influence decision points in the adversary's operation resulting in action, or inaction, favorable to the deceiver's objectives. Alteration of the adversary's perception of the situation at key decision points may result in suboptimal course of action selection or termination of the entire operation [4: 3]. For this reason, deception is most effective if accomplished early to exploit the adversary's unfamiliarity with the defender [13: 151]. Knowledge of the adversary's motives and objectives is also essential [4: 19].

Considerations

Deception planning can be complex and resource intensive – many factors must be considered. Of these, secrecy, unintended consequences, counter-deception and deterrence have bearing on this research and merit brief discussion.

Secrecy is a fundamental requirement for deception to be effective. Adversaries are less likely to behave as desired if aware of a deception. However, since deception is based both on showing and hiding, complete secrecy is undesirable. For each deception, an understanding of what must be kept secret and what should be revealed is essential [4: 15].

All deceptions have the potential for unintended consequences. For example, when one deception solution developed to limit the effectiveness of network scanning was deployed for the first time, it incapacitated the defensive tools used to detect vulnerabilities [4: 22-23].

Deception can induce counter-deception by the attacker. For instance, techniques for detecting Sebek – an essential component of the Honeynet Project's honeypot

architecture – have been developed to defeat its concealment measures. However, if the resource burden to overcome a deception is significant enough, the counter-deception measure may be to the deceiver's advantage [13: 152].

Advertising the use of high quality deceptions could have a deterrent effect on potential adversaries, especially less skilled and/or poorly resourced one. Awareness that high quality deceptions are in widespread use may (a) cause potential attackers to choose softer targets for fear they may be unable to differentiate deceptions from non-deceptions; and/or (b) because they believe the cost of differentiating them is unacceptable [4: 16].

Deception in Network Defense

Through observation of red teaming – friendly forces posing as adversaries to assess defensive tactics, techniques and procedures – and penetration testing, researchers have reached several general conclusions on the behavior of people who use computers for network attack. First, they form expectations based on experience with their tools and targets. Second, they tend to trust results provided by their tools unless they deviate too far from expectations. Third, they use computers primarily to automate manual processes and not to augment reasoning [4: 55]. Additional empirical research is necessary to validate these observations, but they bode well for the potential of defensive network deception.

Defensive network deception is directed at the intellect and skills of the adversaries behind attacks, not the tools they employ. That is, while successful deceptions cause systems to perform differently due to their inability to differentiate deception from reality, ultimately it is the adversary who must be influenced to achieve the desired objective [4: 32]. Network defenders thus employ deception to "mislead attackers into a

predictable course of action, or inaction, which can be exploited or otherwise used to advantage" [19: 26]. Example techniques developed include (a) altering the signature of a computer so an adversary's tools identify the incorrect operating system, resulting in execution of an ineffective attack; and (b) simulating a vulnerable database to luring adversaries from production systems into an environment where their actions are monitored and recorded.

Despite the potential of defensive network deception, it appears to be limited in use – most likely due to the aforementioned resource requirements. Network attackers routinely use deception, most frequently as identity deception and concealment [13: 151]. However, large scale use of sophisticated deceptions appears limited in network defense. Instead, minor deceptions – such a false system responses – appear to be the norm. For example, providing non-descript responses to failed authentication attempts [19: 26].

Dynamic Network Reconfiguration

Based on the strategy of continuous unpredictable change, this defense operates by maintaining unpredictability for the attacker. To succeed defenders must be able to continue operations, while attackers are unable to perform reconnaissance and execute effective attack within a single configuration change cycle. Timeliness of the change is a key consideration – too high a rate of change may cause performance degradations, while too low a rate may provide attackers sufficient time to strike successfully. Examples of this strategy include address hopping, unpredictable server selection, and unpredictable network route selection [3: 9].

A form of obfuscation, address hopping is an active defense tactic that dynamically changes a computer's identity with the dual objective of hiding its real identity and

confusing the attacker during reconnaissance [3: 6]. It has been the subject of numerous journal articles in the last decade – to include an English language article by Chinese researchers [18] – and its effectiveness has also been demonstrated in several experiments. Unpredictable server selection employs a pool of servers and dynamically changes the server selected to answer client requests. Unpredictable route selection dynamically alters the route packets take to reach their intended destination – selection is thus no longer based on routing efficiency. Of the three, address hopping appears to have the greatest potential for obstructing network reconnaissance efforts.

III. Technologies of Interest

Three technologies were identified early in this research project as highly relevant to the AFCYBER (P) concept and are discussed in this section – address hopping, honeypots and network telescopes. Many other technologies are relevant and warrant investigation, but they are beyond the scope of this effort and left to follow-on research.

Address Hopping

Overview

Address hopping – also called network address space randomization [2] and address obfuscation [1] – is an active defense tactic that dynamically changes a computer's network identity with the dual objective of hiding its real identity and confusing the attacker during reconnaissance [3: 6]. It appears researchers began exploring this defensive tactic in earnest in the early 1990's, concentrating on TCP/IP and Ethernet networks. Sandia Labs researchers identified the following protocol fields as having the potential for use in the obfuscation process: (a) MAC address, (b) IP address, (c) IP Type of Service field, (d) TCP port, (e) TCP sequence number, (f) TCP window size, and (g) UDP port [9: 12]. Of these, only the host portion of the IP address, the TCP/UDP port number, and the MAC address were employed in the majority of literature uncovered. They appear to be the most effective identity information to alter dynamically, especially in combination.

In TCP/IP communications, all traffic exchanged between two parties contains a source and destination address pair consisting of an IP address and Port number. Similarly, Ethernet-based local area networks use the network card's MAC address to identify source and destination points. Port hopping continuously replaces the source and destination port with pseudo-randomly picked numbers. IP hopping pseudo-randomly changes the IP address part and MAC address hopping pseudo-randomly changes the MAC address. Once altered by the address hopping software, traffic intercepted by attackers will reveal random addresses valid only for a small time interval. This enables obfuscation of the true identity of machines and services from an OSI model layer 2 (data link) and 3 (network) perspective; and, is sufficient to defeat many network-level reconnaissance tools – such as NMAP [6: 2] – which rely on an accurate host mapping to correctly correlate port and operating system intelligence [9: 29]. To succeed, attackers must discover the current address pair and execute the attack within the current refresh cycle. Additionally, the probability of intruder detection is increased since attackers risk operating against "stale" addresses and thus raising alerts [3: 6; 17: 38].

There are a variety of ways to implement this defense [9: 1], but most appear derived from Network Address Translation (NAT) or Dynamic Host Configuration Protocol (DHCP) technology. The two best-documented solutions discovered in this research effort were Dynamic Network Address Translation (DYNAT) and Applications Which Participate in their Own Defense (APOD). Both were developed under DARPA's Information Assurance Program, but differ in objective. DYNAT is a dedicated address hopping solutions to protect communications between network clients and servers.

APOD is a defensive system, centered on applications, which includes address hopping capabilities. The DYNAT solution – based on NAT technology – is covered throughout the remainder of this report due to the thoroughness and quality of technical documents discovered for it.

Effectiveness

In 2001, DARPA conducted a series of experiments to assess the effectiveness of address hopping with the assistance of a red team from Sandia Labs. DYNAT was employed to provide the address hopping capability [6: 3] with two- and five-minute remapping intervals [6: 6]. The objective was to compare attacker work load in the network discovery phase for both static and dynamic reconfiguration network architectures [6: 3].

Reduced Effectiveness of Traffic Analysis

In the first series of experiments, a simulated environment was built modeling a deployed military unit traversing a public network to access remote resources. A class-C addressing scheme was used and common services such as Domain Name System (DNS), e-mail, TELNET, and File Transfer Protocol (FTP) were running. Additionally, several web-enabled databases containing maps, logistics data, intelligence data, and personnel data were running. Users could view critical planning documents containing keywords – such as “weapon”, “sortie”, “target”, and “bomb” – but access was restricted with user ids and passwords. Automated traffic generators were used to simulate routine traffic and encryption was used to prevent sniffing of clear text web traffic for the keywords. However, to isolate the effectiveness of address hopping alone, no Intrusion Detection

Systems (IDS) were used. The adversary's goal was to stealthily map the defender's network and identify the critical servers by passively sniffing network traffic from the simulated public Internet [6: 3-4]. Eight experimental runs were conducted with the following configurations:

- Runs 1, 2, and 3: DYNAT was disabled to baseline adversary work load on a static network.
- Runs 4 and 5: DYNAT was enabled, but the adversary was given no intelligence on the existence or operation of the dynamic network reconfiguration mechanism. This measured the effectiveness of the mechanism against an adversary with no prior knowledge.
- Run 6: DYNAT was enabled and the adversary was given some intelligence of the defense. This measured the effect of an adversary gaining some knowledge of the mechanism – the most likely scenario.
- Run 7: DYNAT was enabled and the adversary was given full intelligence on the mechanism, to include the source code. This measured the effect of an adversary gaining full knowledge of the mechanism – a likely scenario when faced with a skilled, well-resource adversary.
- Run 8: DYNAT was enabled, the adversary was given full intelligence, and encryption disabled to gain insight on the importance of data encryption [6: 4-5].

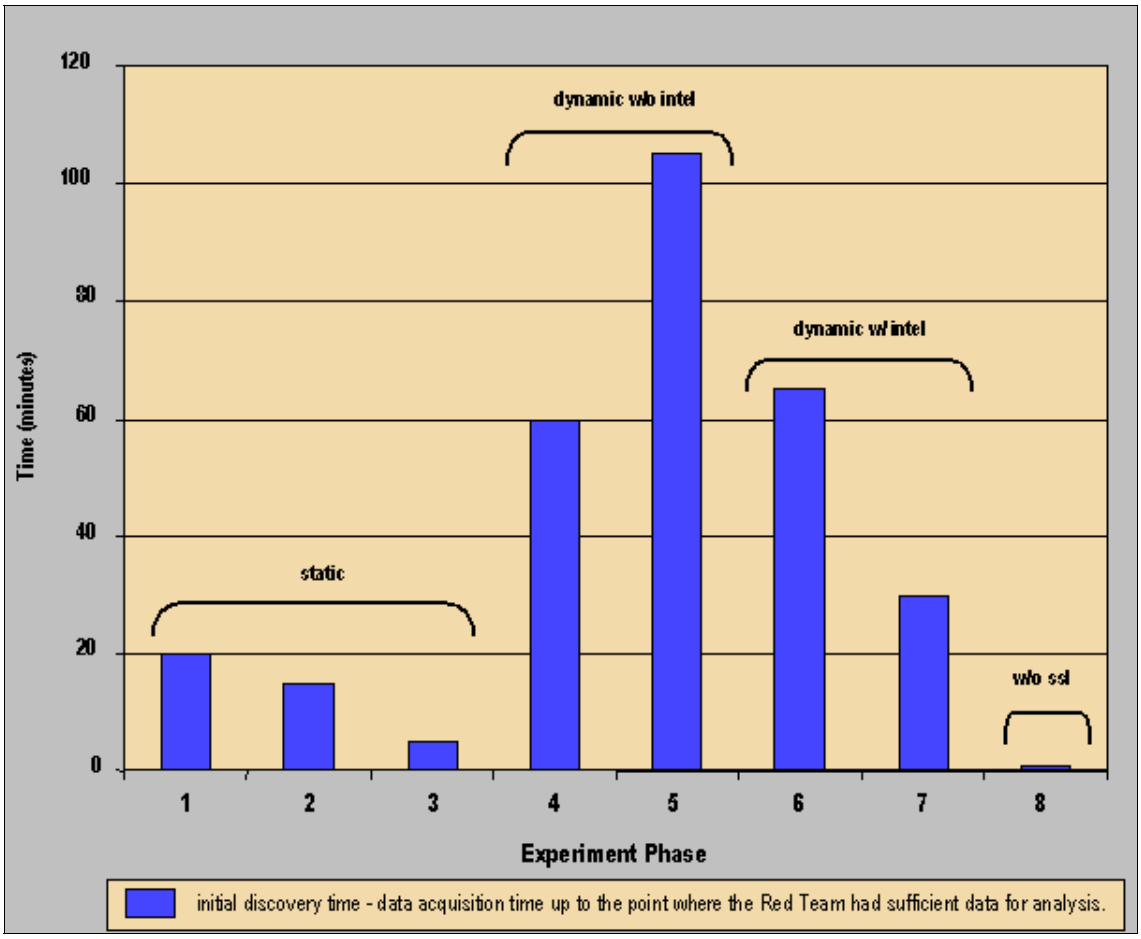

Figure 2: DARPA Experimental Results [6]

The experimental results summarized in Figure 2 (above) provide strong evidence supporting the effectiveness of address hopping. It did make the mapping and discovery efforts more difficult for the adversary. Without knowledge of the hopping defense (runs 4 & 5), the red team was completely unsuccessful. Even with intelligence on the remapping technique (runs 6 & 7), the red team was not able to attain their objectives in a timely fashion. The increased work factor was estimated in the range of 2:1 to 4:1 overall [6: 5].

To the casual observer, address hopping protected traffic would likely reveal little of interest as there are not consistent threads of communication to follow between hosts.

For a determined adversary, the experiments demonstrated a substantial data collection and subsequent analysis effort would be required to obtain useful information. This would increase loiter time on the network and thus increase the risk of detection [6: 5].

However, in the absence of encryption, skilled adversaries armed with knowledge of their target would likely fair better. Experimental Run 8 highlighted this vividly. With experience gained from the previous seven experimental runs and full intelligence on the mechanism, the lack of encryption was devastating. Aware of the address hopping defense and armed with knowledge on the target, the red team was able to quickly isolate the critical servers by searching the clear text traffic for the keywords. Complete mapping of the network was still complicated by the address hopping defense, but identification of the critical servers was not [6: 5].

Reduced Effectiveness of Network Scanning and Mapping

In the second series of experiments, the simulation network was modified to model a fixed-base military network complete with Demilitarized Zone and IDS. A class-B address scheme was used and DYNAT provided the address hopping capability. Again the red team operated from an external network location with the objective of identifying the blue force critical servers. Instead of passive sniffing and traffic analysis, the red team this time used NMAP – currently the most widely employed open source network mapping tool – to conduct active network reconnaissance [6: 8].

As is typical for this type of tool, the scanning process begins with live host discovery. NMAP sends an ACK packet to each IP address in the target address space, expecting a RESET packet in return from live hosts. IP addresses are selected randomly

to reduce the probability of alerting an IDS. Once a live host is identified, NMAP then begins the SYN scan. In a general sweep, TCP SYN packets are sent to each port on the target system, again in random order. However, the attacker can target a specific port or port range. NMAP continues alternating between ACK and SYN scans until the entire target address space is examined [6: 8].

The red team began by using a 'low and slow' approach to minimize the probability of detection. The entire address space was scanned targeting only port 80 – standard port for the HTTP service, commonly thought of as web service – but they were unable to identify any live hosts. The randomization of host IP addresses and port number pairings prevented NMAP from accurately mapping ports to hosts. It's important to note the tool perceived it was consistently and correctly trying port 80 over the entire address space [6: 8], enhancing the effectiveness of the deception. By receiving normal feedback from the tool – instead of an error message – the adversary was forced to remain on the network longer, increasing the probability of detection.

While the red team's network reconnaissance efforts were defeated in this time-limited experiment, address hopping will likely not defeat well-resourced, skilled adversaries – such as nation-states and state-sponsored organizations – in the wild. In subsequent experiments conducted by Singapore's Defense Science Organization, adversaries were able to successfully map the target network. However, the results further validated address hopping's ability to increase the resources and risk required to obtain useful reconnaissance at a reasonable cost to defenders. To succeed, the adversary was forced to use more time consuming and aggressive techniques resulting in a higher probability of detection [6:8 & 10; 9: 53].

Increased Probability of Intruder Detection

Intrusion Detection Systems collect information from a variety of system and network sources, and then analyze it for signs of external intrusion and internal misuse. They're generally divided into two types: host-based and network-based systems. Host-based IDS are typically software resident on the protected host. They monitor for anomalies during user operations – such as improper file manipulations, kernel calls, or application access violations. Network-based IDS (NIDS) are typically dedicated systems. They monitor all network traffic, comparing it against attack 'signatures' – known malevolent activity patterns – to identify potential intruders and alert defenders. While effective against unskilled attackers, these systems often prove inadequate against the stealthier network reconnaissance activities of skilled attackers. These adversaries proceed slowly and attempt to hide in the network's background radiation – the clutter of normal traffic – while performing their reconnaissance [9: 28-29].

However, if defenders are using address hopping, then traffic entering the network addressed to non-existent hosts or ports should cause suspicion. Statistically, most intruder traffic will be directed to "stale", e.g. inactive, host:port pairs, and are therefore easily distinguishable from legitimate traffic [6: 6]. The effectiveness of NIDS can thus be improved by integration with the address hopping system. Comparison of suspect traffic with stored address hopping history would enable correlation with a specific target host. This synergistic defense strategy was evaluated in tandem with the second series of DARPA experiments noted above and proved successful at increasing both the quantity and quality of alerts over a typical IDS [6: 8-10]. As a further step, the researchers noted

the suspect traffic could be also be redirected into a honeypot for covert observation and potential attribution [9: 29].

Technical Overview

The DYNAT solution is provided in this section but generalizes well to address hopping solutions in general. As previously noted, address hopping solutions operate by obfuscating host identity information. They can be implemented purely as software, in dedicated hardware or a combination of both [9: 6]. Regardless of form, addressing information originating from a sending client is translated prior to transmission to the receiver. The receiver then reverses the translation to obtain the true host identity information. The translation algorithm on both ends uses a cryptographic engine with a shared secret initial seed value that varies with time [6: 2]. A process control mechanism is also required to coordinate the encoding change timing [3: 6].

Process Control Mechanisms

Three general process control mechanisms are described: time base (synchronous), time base (polling) and packet/frame based [9: 2]. DYNAT uses time-based synchronization.

Both the synchronous and polling approaches use a clock to determine when an encoding change is needed. Timing must also be synchronized across all participating nodes – for instance by referencing a master network timing source – to ensure synchronized switchover. Importantly, timing is integrated into the cryptographic key in the synchronous approach, whereas polling utilizes it solely to coordinate switchover timing. The synchronous approach is also distributed, whereas polling uses a controller

node responsible for coordinating the encoding changes. The packet/frame approaches uses agreement on pre-coordinated values, instead of time, to synchronize encoding changes. The rate of change is based on the number of packets/frames in each start-up session and alterations require re-coordination [9: 2-3].

Architecture Considerations

Addressing hopping devices can be integrated at a variety of locations in the network architecture based on the level of capability required and the need to preserve the functionality of other services. While not all inclusive, five likely use cases described by Sandia National Labs researchers are provided as examples. They are summarized in Table 1, followed by an overview of their architectures. Design considerations, such as the protection afforded by placement of the hopping mechanism and its impact to common security devices, are also presented [9: 6-11].

Table 1: Sample Address Hopping Use Cases

USE CASE	SCENARIO	CONSIDERATIONS
LAN Segment with Distant Server	Servers not local; LAN segment with clients trusted but link to servers untrusted	Router packet filtering functionality; Intrusion Detection Systems impact
LAN Segment with Local Server	Servers local; LAN untrusted	Intrusion Detection Systems impact
Local Router to Local Router	Servers separated by directly connected routers; LAN segments trusted, but interlink untrusted	Hopping only across router interlink
Gateway to Gateway	Servers separated by untrusted WAN; LAN segments trusted, but WAN untrusted	Hopping only between gateways; VPN/Firewall impact
LAN Segment to LAN Segment	Servers separated by untrusted WAN; LAN segments untrusted	Intrusion Detection Systems impact; VPN/Firewall impact

LAN Segment with Distant Server

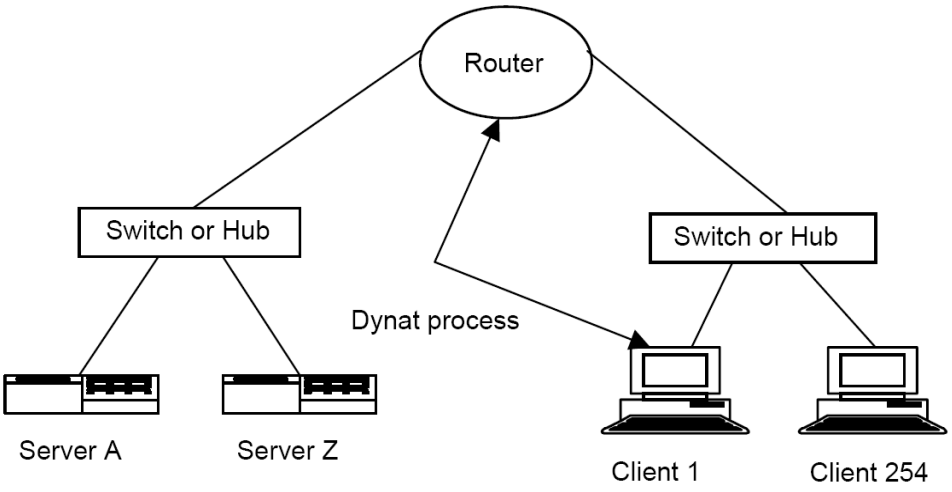

Figure 3: LAN Segment with Distant Server [9]

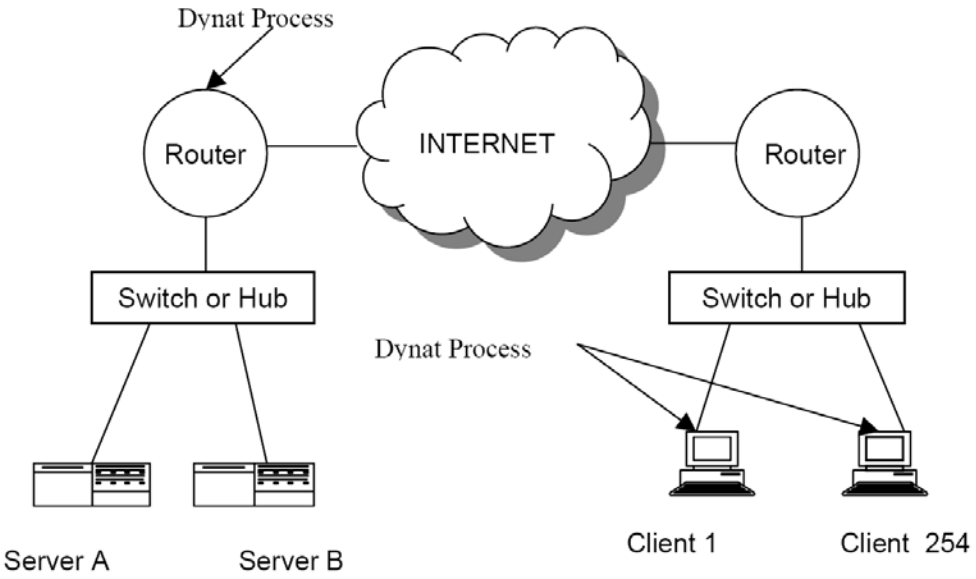

Figure 4: LAN Segment with Distant Server [9]

This use case describes a switched LAN segment with all client machines and the router running the address hopping process. No servers are present on the local LAN segment. All client/server interactions cross a routed interface leading to another local

LAN segment (Figure 3) or through the Internet for a remote connection to a distant server (Figure 4).

The address hopping process can be integrated with the router or implemented on a stand-alone device. If a stand-alone solution is employed, it must be installed after the router to preserve packet filtering functionality. If a NIDS is employed it must also participate in the address hopping process or it will be rendered useless [9: 17].

LAN Segment with Local Server

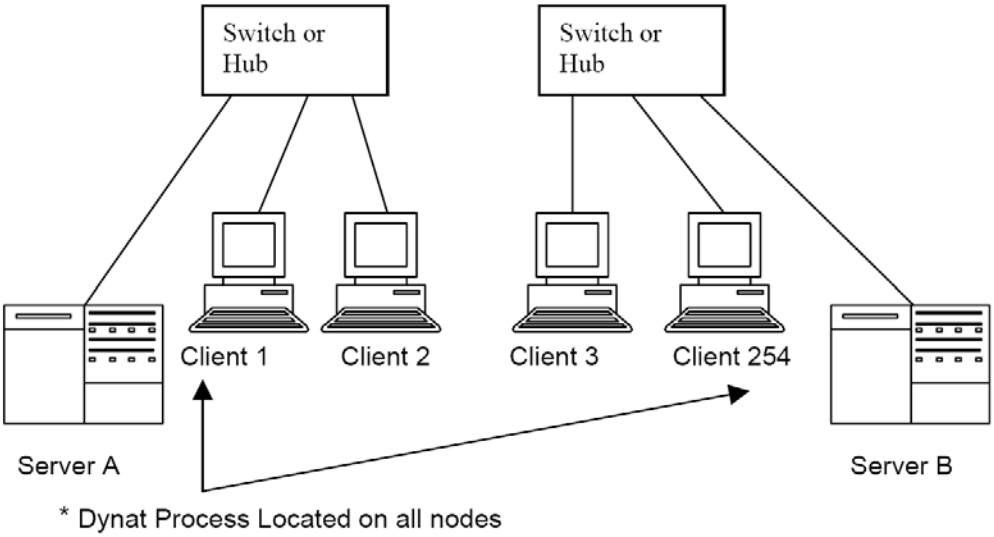

Figure 5: LAN Segment with Local Server [9]

This use case is the simplest for address hopping. Clients and servers reside on the same LAN segment and every host is running an address hopping process (Figure 5). There are no routers or gateways and all nodes are connected through switches or hubs. Firewall and proxy server function are unaffected by this implementation, since all hopping traffic is localized. However, NIDS devices must still be integrated into the address hopping process to function properly [9: 17].

Local Router to Local Router

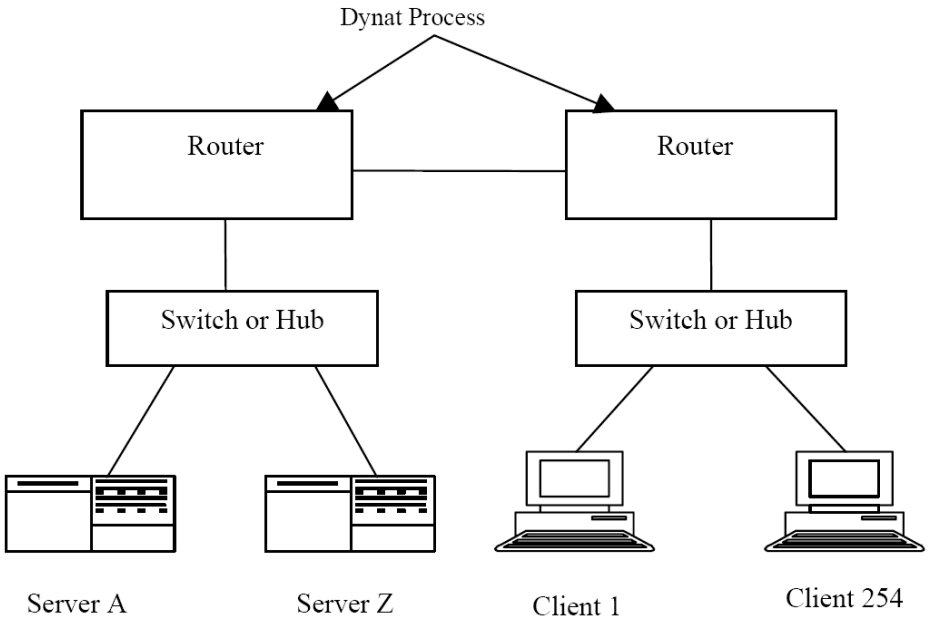

Figure 6: Local Router to Local Router [9]

In this use case, clients and servers are located on different LAN segments connected via directly linked routers (Figure 6). Neither clients nor servers host the address hopping mechanism. Instead it's integrated in the routers or with stand-alone devices at the routers. This technique would be employed where the LAN segments are considered secure, but the interconnecting router link may be exposed to an outside or unprotected enterprise or campus network environment. Similar to the first use case, to preserve packet filtering the address hopping mechanism must be on the router or installed on the interconnecting router link. If this functionality is not used, the device can be installed before or after the routers with no difference. As long as the subnet mask – used on the local segment to differentiate network ID and host ID – is the same one applied at the

router interface, the router will be able to route the network ID portion of the IP address [9: 16].

Gateway to Gateway

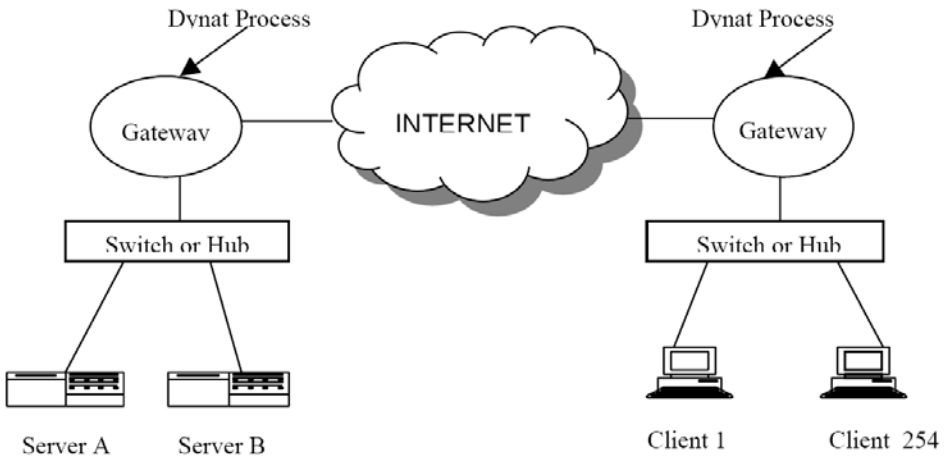

Figure 7: Gateway to Gateway [9]

Figure 8: Address Hopping Mechanism Placement [9]

This use case is similar in architecture to the previous, but the routers in this case connect through an unsecure public communication medium – for example, the Internet – and were thus now referred to as gateways by the researchers (Figure 7). Again the LAN segments are considered secure, and only end node address assignments crossing the unsecured public communication medium are protected.

If Virtual Private Network (VPN) or Firewalls systems are employed, address hopping must be implemented on the link between these devices (Figure 8). VPNs require that each participating gateway have a static, authenticated network identity at each other's node interface. Rudimentary firewalls track state information about originating connections, while application-level firewalls add a proxy function. None of these systems will function properly if the network information needed is continuously being changed [9: 16].

LAN Segment to LAN Segment

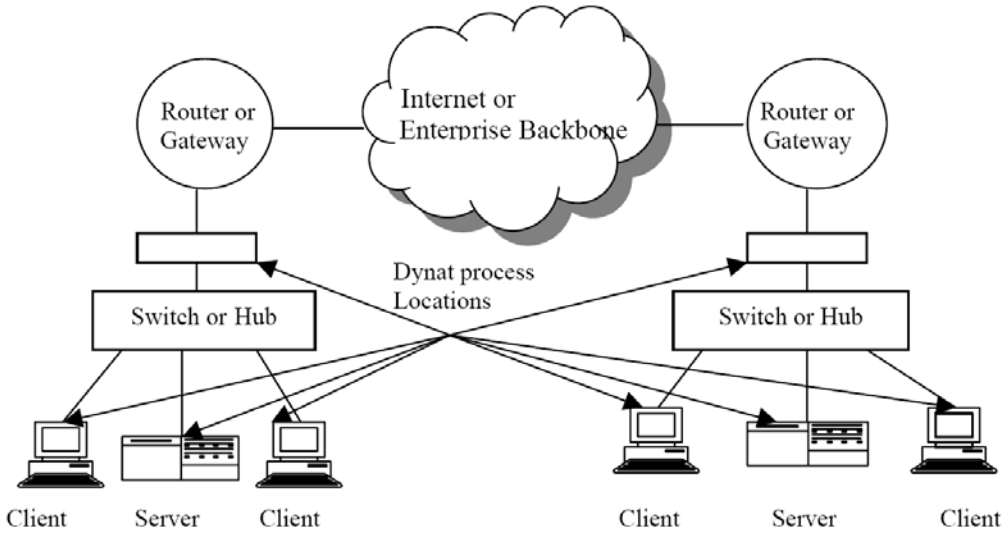

Figure 9: LAN Segment to LAN Segment [9]

This final use case is the most complex to implement. As in the previous use case, gateways are connected via an unsecure public communication medium. However, the LAN segments are also deemed unsecure and thus address hopping is employed on all local hosts as well (Figure 9).

NIDS integration with the address hopping mechanism is once again necessary. Additionally, the VPN and firewall architectural requirement of the previous two use cases remain. A clear technical solution wasn't provided for this use case. However, it is assumed that to avoid disruption of firewall and VPN capabilities, address obfuscation implemented at the host-level is reversed prior to passing outbound through these devices and then reapplied. At the receiving end translation again occurs prior to these devices as per the previous use case [9: 17].

Integration with Encryption Services

As noted earlier, address hopping and encryption are mutually supporting defense technologies. Internet Protocol Security (IPSec) in particular is an attractive companion to IP/port address hopping due to its flexibility, cryptographic strength and ease of employment. IPSec was designed to encrypt and authenticate TCP/IP communications without requiring special application modifications – as with the Secure Sockets Layer solution. Originally designed as an integral component of Internet Protocol Version 6 (IPv6), it has since been implemented in IPv4 as well.

Two operating modes are provided, tunnel and transport. In transport mode, communication can be controlled and protected at the port and protocol level. In tunnel mode, all communication between gateways is encapsulated and protected. Transport mode offers greater access control granularity, but creates a higher administrative burden which must considered before mode selection. Additionally, if authentication is used it can't be used with address hopping as this invalidates the authentication hash value. Tunnel mode is most frequently employed to protect traffic between gateways. It provides greater anonymity, and resistance to both network mapping and traffic analysis,

since all traffic is encapsulated and encrypted with only the gateway IP addresses visible within the protected communication stream.

While there is overlap in the protection services they provide – access control, network scanning resistance, anonymity, and traffic analysis resistance [9: 21-23] – combined they can provide a stronger defense. The interaction between address hopping and IPSec can occur as both: (a) Address Hopping over IPSec, and (b) IPSec over Address Hopping [9: 21-23]. Of the two, the second appears the better combination.

In both combinations, researchers found that IPSec's ability to hide payload and header information through encryption increased the anonymity and traffic analysis resistance provided by address hopping. However, in the first combination IPSec's security association and key management features reveal identity information, thus compromising the anonymity and traffic analysis resistance [9: 27].

Performance and Interoperability Considerations

Security technologies and practices often introduce measurable degree of hindrance or degradation [9: 12], and address hopping isn't an exception. A detailed cost-benefit analysis should be conducted for every potential fielding. Despite its defense capabilities, address hopping may not be appropriate for all situations as it can interfere with numerous applications and may degrade network performance.

Network Performance Degradation

Serious degradation can occur when MAC address hopping is employed due to the increased number of Address Resolution Protocol (ARP) messages generated and the

memory limitation of Ethernet switches. All hosts on a network segment have a unique MAC address, which is tied to the connecting port on its servicing Ethernet switch. Each switch maintains a table mapping the source MAC address of each Ethernet frame received and its associated port. This mapping allows the switch to direct Ethernet frames to their proper destination [9: 12-13].

When MAC address hopping is employed, an ARP message is generated each time the address is changed so the servicing Ethernet switch can update its table and distant hosts can update their ARP caches. This process produces a three-fold performance hit: (a) All hosts must process the ARP message, (b) additional entries are added into the Ethernet switch tables, and (c) valuable port and bandwidth resources are consumed. The overall performance impact on the network depends on the change rate and number of participating hosts. Critically, some Ethernet switches lock up when their available table memory is exceeded [9: 13-14], resulting in a self-imposed denial of service. However, including switches in the hopping architecture and/or increasing their memory capacity are potential solutions worth exploring.

Application Interoperability Issues

Unless specifically designed to account for them, address hopping mechanisms can cause numerous application interoperability issues when transmitting or receiving data from outside a local segment [9: 15]. For instance, applications that contain IP address and port information in the data payload will not function correctly. This happens because the original IP address or port number is changed, but the information in the data payload, referencing the original header, remains unchanged. Use of application-level

gateways to correct the information in the data payload could address the problem, but would introduce additional complexity and wouldn't work with applications employing payload encryption [9: 14].

Additionally, protocols which employ both a control and data port – such as File Transfer Protocol, H323, Session Initiation Protocol and Real Time Session Protocol – will be disrupted. These applications exchanged address and port control information before establishing a data session. Unless the address hopping mechanism is designed to cope with these interdependent sessions, they will fail [9: 14].

Finally, problems may arise when a peer-to-peer application – such as Instant Messaging and IP Telephony – attempts to create a session between hosts in both the address hopping protected enclave and an unsecured public address space. Establishing a connection will only be possible when the session originates from within the address hopping protected enclave [9: 14].

Incompatibility with MAC Port Locking

MAC port locking is a security feature on Ethernet switches that allows MAC addresses to be associated with specific ports. When used, data is only accepted from a port when the source MAC address field matches the assigned MAC address. If MAC address hopping is employed, MAC port locking must be deactivated to avoid a self-imposed denial of service [9: 17].

Defeating Address Hopping

The address hopping defense can be attacked directly and indirectly. The direct approach involves monitoring traffic and attempting to extrapolate the hopping pattern.

It would likely be resource intensive and prone to failure, especially if a short hopping interval, robust randomization algorithm and encryption were employed. Acquiring detailed intelligence on the hopping mechanism would simplify the process, but provide no guarantee of success. Obtaining a working device, plus the randomization algorithm initialization values and keying material would defeat the defense, but only until the initialization values and keying material were changed.

The indirect approach offers more promise. Given the proven susceptibility of people to computer-based deception, users could be targeting with 'spear phishing', Trojan software or other deceptions with the objective of installing malware on a trusted host. Once compromised, the system would enable participating in the address hopping defense. The potential of unwittingly including undiscovered compromised hosts during address hopping implementation is an additional concern.

Honeypots

Overview

Honeypots are a form of decoy. The Honeynet Project defines them as "information systems whose value lies in their unauthorized or illicit use". They're designed to duplicate an application or system as closely as possible with the objective of deceiving intruders into interacting with them. All activity on them is monitored, logged and captured. Additionally, they include features to limit their effectiveness as an attack platform if compromised. For this reason, they're frequently used by researchers to gather data on network attack tactics and tools. They can also be employed to decoy intruders and gather intelligence about vulnerabilities and compromises on operational

networks [15]. However, they don't appear to be used for this purpose on a large-scale – probably due to the time and skill required to produce the high-quality honeypots needed to deceive skilled adversaries.

There is no single standard honeypot solution. Specific solutions are tailored to the requirements of the situation. While using real systems is possible, virtualized solutions – such a VMware – are often used due to the ease with which they can be restored to their baseline configuration. Honeypots fall into two general categories – low and high interaction [15].

Low-interaction honeypots emulate services – web, email, FTP, etc – rather than complete systems. Interaction with the operating system is not emulated. Due to their limited capabilities, these honeypots are easily detected by knowledgeable adversaries [13: 152]. Honeyd is one example of a low-interaction honeypot. It's used to monitor the unused IP address space on a network. Anytime it sees a connection attempt to this region, it intercepts the connection and pretends to be the victim service [15].

High-interaction honeypots masquerade as a full system, emulating all service provided plus the underlying operating system. Alternatively, a real operating system installation with added monitoring features can be employed. Since interaction with a full operating system is possible, defeating the deception is more difficult [15]. However, the risk of compromise is also greater. Honeynets are a form of high-interaction honeypot which emulate an entire network. They may be comprised entirely of decoy systems or include productions systems as well [7: 31].

Detection

As decoys, Honeypots are effective only as long as their masquerade is maintained. Once reveled, intruders typically cease operations on the system. Specific concealment and discovery techniques continue to change as developers and attackers engage in an ongoing arms race [12: 35]. However, the data control and capture measures integral to honeypots are the underlying cause of their vulnerability to detection.

Due to the threat of compromise, data control measures are used to protecting non-honeynet systems that an attacker might target from a compromised honeypot. A common tactic is limiting the number of outgoing connections or bandwidth usage allowed. Alternatively, outbound malicious traffic can be sanitized. In the first case, attackers can detect these measures when attempts to exceed the thresholds fail. In the second, send a malicious packet to a visible external to a machine under the attacker's control reveals the alteration [7: 9-10].

The data capture – logging, monitoring, capturing – facilities may also be detectable due to system modification they makes, activity traces they leaves or system response delays they cause. For example, despite a determined concealment effort, attackers discovered how to detect the Honeynet Project's Sebek software by the system anomalies it creates [7: 9-10].

The presence of honeypots can also be inferred by identification of the host system as a virtual machine. If the attack believes virtual machines are unlikely to be used as production system, then virtual machines detected are more likely to be honeypots. They will be avoided or scrutinized closely before entry. However, the use of virtual machines as production systems is rapidly increasing, especially with servers – the highest value

targets for many network attackers. If this trend continues, the usefulness of virtual machine identification for honeypot detection will be greatly diminished.

Fake Honeypots

An interesting idea based on the deterrence is the concept of "fake honeypots". Network attackers avoid honeypots. They understand the threat they present – revealing tactics, techniques and tools – and their limited effectiveness as an attack platform if compromised. This suggests that pretending to be a honeypot might help deter attackers [12: 25].

Low cost and easy to implement, fake honeypots could be a force multiplier. Used in tandem with high-quality honeypots, they could decrease an adversary's confidence and increase the perceived operational risks. Additionally, poorly resourced attackers might be convinced to seek easier targets [12: 27].

Network Telescopes

Network telescopes are a type of sensor used to detect overt large-scale malicious activity – such as denial of service attacks, network scanning and worm propagation – on the Internet. They're deployed in regions of routable, but unused IP address space (e.g. dark IP space) where legitimate traffic shouldn't appear. They operate by passively monitoring for the arrival of unexpected traffic [10: 1], in particular the signs of scanning and backscatter from malicious activity. While the traffic of interest is malicious, the network's 'background radiation' – garbage traffic that serves no purpose, such as damaged or improperly routed packets – must first be characterized and accounted for to

isolate it [11]. Unfortunately, this isn't a one-time effort. Malicious actors seeking stealth work continuously to make their activities appear innocuous.

Due to their associated analysis burden and inability to detect subtle activity, network telescopes are used primarily by researcher and large Internet security firms for trend analysis and monitoring/early warning of major activity.

However, while not suited for large-scale deployment, they remain of interest. A limited number of large fixed telescopes could be augmented by a flight of small mobile telescope. Using dynamic network reconfiguration technology, these small telescopes could sample broad regions of address space – operating like mobile gap-filler radars in an Integrated Air Defense System.

IV. Conclusion

Findings

The proposed AFYBER (P) strategy has merit. Dynamic network reconfiguration and decoys are effective means to attack adversary network reconnaissance efforts and improve network defense. They generate unpredictability and uncertainty, increasing resource costs and risks in the adversary's IPB process. Potential effects include deterring resource-constrained or risk-averse adversaries, inhibiting maturation of the attack planning process (thus preventing attacks), and degrading the quality of attacks.

Individual technologies supporting this strategy exist today and have demonstrated effectiveness. Address hopping reduces the effectiveness of network scanning and mapping, and increases the probability of intruder detection. IPSec improves anonymity and traffic analysis resistance. Network telescopes provide event detection and an early warning capability. Honeypots divert attention from production systems, increase the probability of intruder detection, and gather intelligence on attack methods.

Additionally, some of these technologies appear mutually supportive, creating synergistic effects when combined. For instance, the effectiveness of detection techniques limits the useful lifespan of honeypots against skilled adversaries. Address hopping may offset this by continuously changing a honeypot's identity, thus necessitating retesting. This may increase the deterrence effect; and reduce the costs associated with producing and managing the high quality honeypots needed to deceive skilled adversaries.

While some technologies are mature and could be implemented relatively quickly, caution is advised and further study recommended to avoid unintended consequences. The dynamic network reconfiguration technologies – address hopping, unpredictable server selection, unpredictable network route selection – in particular could be very disruptive. Extensive analysis and engineering is needed to determine which technologies are appropriate, how and where to integrate them into our networks and how to employ them most effectively.

Future Research

Manpower Requirements and Organizational Impacts

Effective employment of the AFCYBER (P) strategy requires skilled specialists in the associated technologies, traffic analysis, deception planning, TTPs and more. Research is necessary to determine the manpower and skill sets required, as well as the organizational implications. For example, centralized or regionalized operation of honeypots seems appropriate due to their development, maintenance and analysis burdens. It seems appropriate for network telescopes as well. However, can either be effectively monitored and controlled remotely in numbers? Should they be organized as new units or integrated into existing ones, such as the INOSCs or network warfare squadrons?

Network Maneuver

Dynamically moving our networks around in IP address space would be highly disruptive to the routing system in place today, especially if a rapid rate of reconfiguration was used. Unlike address hopping using the host-portion of the IP address, mature solutions weren't found. Research is needed to determine how to minimize router infrastructure disruption and its effectiveness at varying change

frequencies. It's achievable, but introduces more complexity and unlike host address hopping, the effects aren't localized. However, these are engineering problems.

The major obstacles to implementing this are likely not technical, but rather process, policy and resistance to change. For instance, the DoD IP management process seems relatively rigid and ill-suited for a dynamically changing network. Research on the potential policy and process obstacles in necessary. Additionally, a wide range of actors spread across multiple organizations, many not Air Force, would likely be impacted – resistance to change will be factor. Identifying and accounting for these actors' concerns will be necessary to 'sell' the strategy.

Network Traffic Generators

Network traffic generators, like Lincoln Laboratory's LARIAT, could be used to increase the realism of honeynets by enabling them to mimic the 'emanations' of operational networks. These honeypots could then pop-up in 'stale' addresses vacated by operational networks (hopping through our address space) or at alternate locations to further confuse observers. Accurate characterization of each network's 'emanations' would be key. These signatures would also likely evolve over time, so determining that rate of change and periodically update them would be required. Though less robust, generic signatures could also be developed. However, if adversaries can uniquely identify networks at a distance by other means, this deception would fail. Success is achieved if the adversary must enter the network to recognize the deception or fails to recognize it at all.

Research is needed to determine if this concept has potential and is achievable. Can we characterize and mimic a specific network's 'signature'? Do they have 'signatures'? And critically …. can they be uniquely identified at a distance by other means?

Bibliography

1. Antonatos, S., & Anagnostakis, K. "TAO: Protecting Against Hitlist Worms Using Transparent Address Obfuscation," *10th IFIP Open Conference on Communications and Multimedia Security (CMS '06).* Heraklion, Crete: IFIP, 2006.

2. Antonatos, S., Akritidis, P., Markatos, E., & Anagnostakis, K. "Defending Against Hitlist Worms Using Network Address Space Randomization," *Proceeding of the 3rd ACM Workshop on Rapid Malcode.* Fairfax: ACM, 2005.

3. Atighetchi, M., Pal, P., Webber, F., & Jones, C. "Adaptive Use of Network-Centric Mechanisms in Cyber-Defense," *Second IEEE International Symposium on Network Computing and Applications.* 2003

4. Cohen, Fred, Dave Lambert, Charles Preston, Nina Berry, Corbin Stewart, and Eric Thomas. "A Framework for Deception". *IFIP-TC11*, Computers and Security, submitted 2001. Retrieved from Fred Cohen & Associates: http://all.net/. 2001

5. Grazioli, S., & Wang, A. "Looking Without Seeing: Understanding Unsophisticated Consumers' Success and Failure to Detect Internet Deception," *Proceeding of the 22nd International Conference on Information Systems.* 192-204. 2001.

6. Kewley, D., Lowry, J., Fink, R., & Dean, M. "Dynamic Approaches to Thwart Adversary Intelligence Gathering," *DARPA Information Survivability Conference & Exposition II.* 2001.

7. Krasser, Sven, Julian B. Grizzard, Henry L. Owen, and John G. Levin. "The Use of Honeynets to Increase Computer Network Security and User Awareness," *Journal of Security Education , 1* (2/3): 23-37 (2005)

8. Joint Chiefs of Staff (JCS). *Military Deception*. Joint Publication 3-13.4. Washington: JCS, 13 July 2006

9. Michalski, J., Price, C., Stanton, E., & Lee, E. *Network Security Mechanisms Utilizing Dynamic Network Address Translation.* Albuquerque NM: Sandia National Labs, November 2002 (SAND2002-3613).

10. Moore, D., Shannon, C., Voelker, G., & Savage, S. *Network Telescopes: Technical Report.* San Diago CA: Cooperative Association for Internet Data Analysis, 2003.

11. Pang, R., Yegneswaran, V., Barford, P., Paxson, V., & Peterson, L. "Characteristics of Internet Background Radiation". *Proceedings of ACM Internet Measurement Conference.* Taormina, IT: ACM, 2004.

12. Rowe, N., & Duong B, C. E. “Fake Honeypots: A Defensive Tactic for Cyberspace,” *7th IEEE Workshop on Information Assurance*. 223-230. West Point, 2006.

13. Rowe, N., & Goh, H. “Thwarting Cyber-Attack Reconnaissance with Inconsistency and Deception,” *Proceedings of the 2007 IEEE Workshop on Information Assurance.* West Point NY: IEEE, 2007.

14. Rowe, Neil and Hy S. Rothstein. “Two Taxononmies of Deception for Attacks on Information Systems,” *Journal of Information Warfare , 3* (2): 27-39 (2004).

15. Spitzner, Lance. *Honeypots: Definitions and Values of Honeypots.* Retrieved from Honeypots: Tracking Hackers: http://www.tracking-hackers.com/papers/ honeypots.html. 29 May 2003.

16. Vrijj, A. *Detecting Lies and Deceit: The Psychology of Lying and the Implications for Professsional Practice.* Chichester, UK: Wiley, 2000.

17. Webber, Frank, Partha P. Pal, Michael Atighetchi, Chris Jones, Paul Rubel, Ron Watro, Tom Mitchell, Richard E. Schantz, and Joseph P. Loyall. *Applications That Participate In Their Own Defense (APOD).* Rome NY: Air Force Research Laboratory, May 2003 (AFRL-IF-RS-TR-2003-103).

18. Yang, C. “Port and Address Hopping for Active Cyber-Defense,” *Pacific Asia Workshop on Intelligence and Security Informatics (PAISI).* 295-300. Berlin DE: Springer, 2007.

19. Yuill, Jim, Dorthy Denning, and Fred Feer. “Using Deception to Hide Things from Hackers: Processes, Principles, and Techniques,” *Journal of Information Warfare ,* 5 (3): 26-40 (2006).

20. Rosenau, William. Special Operations Forces and Elusive Enemy Ground Targets. Santa Monica CA: RAND, 2001.

21. Collberg, Christian and Clark Thomborson. “Watermarking, Tamper-Proofing, and Obfuscation – Tools for Software Protection,” *IEEE Transactions on Software Engineering*, 28 (8): 735-746 (2002).

22. Collberg, Christian, Clark Thomborson, and Douglas Low. *A Taxonomy of Obfuscating Transformations*. Technical Report 148, Department of Computer Science, University of Auckland, July 1997. Retreived from http://www.cs.auckland.ac.nz/~collberg/Research/Publications/CollbergThomborson Low97a.

Additional Sources

Apel, Thomas. *Generating Fingerprints of Network Servers and their Use in Honeypots*. Thesis. Aachen University, Aachen DE, 2005.

Athanasiades, Nicholas, Randal Abler, John Levine, Henry Owen, and George Riley. "Intrusion Detection Testing and Benchmarking Methodologies," *First IEEE International Information Assurance Workshop*. Atlanta, GA: Georgia Institute of Technology, 2003.

B. T. Duong, "Comparisons of attacks on honeypots with those on real networks," MS thesis. Naval Postgraduate School, Monterey CA, March 2006.

Cohen, Fred, and Deanna Koike. "Leading Attackers Through Attack Graphs With Deception," *IFIP-TC11*, Computers and Security, submitted 2002. Retrieved from Fred Cohen & Associates: http://all.net/.

Cloud, Donald Wayne Jr. "Integrated Cyber Defenses: Towards Cyber Defense Doctrine," Thesis. Monterey CA: NPS, December 2007.

Department of Defense. *The Role and Status of DoD Red Teaming*. Washington DC: Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, September 2003.

Dornseif, M., T. Holz, and C. Klein, "NoSEBrEaK – attacking honeynets," *Proc. IEEE Workshop on Information Assurance and Security*. West Point, NY: June 2004.

Harrop, W. and G. Armitage. "Greynets: a definition and evaluation of sparsely populated darknets," *In Proceedings of the 2005 ACM SIGCOMM Workshop*. Philadelphia PA: 171–172, August 2005.

Holz, T. and F. Raynal, "Detecting honeypots and other suspicious environments," *Proc. 6th SMC Information Assurance Workshop*. West Point, NY: 29-36. June 2005.

Lee M. Rossey, Robert K. Cunningham, David J. Fried, Jesse C. Rabek, Richard P. Lippmann, Joshua W. Haines, and Marc A. Zissman. *LARIAT: Lincoln Adaptable Real-time Information Assurance Testbed*. IEEE, 2002.

Lippmann, R.P., K.W. Ingols, C. Scott, K. Piwowarski, K.J. Kratkiewiez, M. Artz, and R.K. Cunningham. *Evaluating and Strengthening Enterprise Network Security Using Attack Graphs*. Lexington MA: Lincoln Laboratory, 5 October 2005 (FA8721-05-C-0002).

Moore, David, Colleen Shannon, Doug Brown, Geoffrey M. Voelker, and Stefan Savage. "Inferring Internet Denial-of-Service Activity," *In 10th USENIX Security Symposium*. 2001.

Quist, Danny. “Detecting the Presence of Virtual Machines Using the Local Data Table,” Retrieved from http://www.offensivecomputing.net/.

Rowe, Neil, E. John Custy, and Binh T. Duong. “Defending Cyberspace with Fake Honeypots,” *Journal of Computers*, 2 (2): 25-36 (April 2007)

Rutkowska, Joanna. “Red Pill…or how to detect VMM using (almost) one CPU instruction,” Retrieved from http://invisiblethings.org. November 2004.

Tinnel, Laura S., O. Sami Saydjari, and Dave Ferrell. “Cyberwar Strategy and Tactics: An Analysis of Cyber Goals, Strategies, Tactics, and Techniques,” *Proceedings of the 2002 IEEE Workshop on Information Assurance*. West Point, NY: June 2002.

Whyte, David, P.C. Van Oorschot, and Evangelos Kranakis. “Tracking Darkports for Network Defense,” *Procceding of the 23rd Annual Computer Security Applications Conference.* Miami Beach FL: 161-171, December 2007.

Zou, Cliff Changchun, Don Towsley, and Weibo Gong. *On the Performance of Internet Work Scanning Strategies*. Amherst MA: University of Massachusetts, 2007 (TR-03-CSE-07).

Vita

Major Keith Repik entered the Air Force through the Air Force Reserve Officer Training Corps program at the University of South Carolina and was commissioned in April 1995. His academic degrees include a B.S. in Psychology, M.S. in Computer Science and a M.A. in International Relations.

Maj Repik has served in operational and staff positions at the squadron, wing, joint and major command levels. His duties have included satellite communications, information operations, intelligence support, resource management, and security assistance. He was selected to attend AFIT in 2007 and is currently completing the Cyber Warfare Intermediate Developmental Education program. Upon graduation he will be assigned to the Pentagon.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
06-09-2008	**Master's Thesis**	May 2007 – June 2008

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
DEFEATING ADVERSARY NETWORK INTELLIGENCE EFFORTS WITH ACTIVE CYBER DEFENSE TECHNIQUES	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Keith A. Repik, Maj, USAF	5d. PROJECT NUMBER 08-296
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	8. PERFORMING ORGANIZATION REPORT NUMBER
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	AFIT/ICW/ENG/08-11
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFCYBER (P) Attn: BG (s) Tony A. Buntyn 245 Davis Avenue Barksdale AFB LA 71110-2279 DSN: 781-4861 e-mail: tony.buntyn@barksdale.af.mil	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The purpose of this research was to assess the potential of dynamic network reconfiguration and decoys to defeat adversary network reconnaissance efforts, thereby improving network defense. Specifically, this study sought to determine if the strategy has merit, thus warranting more resource intensive research and engineering studies. The research objective was achieved through a comprehensive literature review and limited technology survey. The key topics examined in the literature review include the network attack process, network defense strategies, deception and continuous unpredictable change. Many candidate technologies were surveyed, but only three identified as high potential were examined in detail: address hopping, honeypots and network telescopes.

The following conclusions were reached: (a) the concept has merit and should be pursued further – dynamic network reconfiguration and decoys have demonstrated effectiveness in controlled experiments; (b) it's achievable in the near term – the essential technologies are available today; and (c) extensive analysis and engineering is needed to determine which technologies are appropriate, how and where to integrate them into our networks and how to employ them most effectively.

15. SUBJECT TERMS
Polymorphic Network Defense, Digital Decoys, Honeypots, Deception, Address Obfuscation, Dark IP Space, Network Telescopes, Network Reconnaissance, IP Address Hopping, Port Hopping, MAC Address Hopping, Dynamic Network Reconfiguration

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Maj Paul Williams, PhD (ENG)
REPORT	ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	**U**	**U**	**UU**	55	(937) 255–3636, ext 7253; e-mail: paul.williams@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18