Disrupting Adversary Decision Logic: An Experience Report

PP Pal[1], NJ Lageman[2,3], NB Soule[1]

[1]*Raytheon BBN Technologies*
Cambridge, Massachusetts, United States

[2]*Amazon*
[3]*(Formerly) BBN*
Boston, Massachusetts, United States

E-mail: partha.pal@raytheon.com; nlageman4@gmail.com; nate.soule@raytheon.com

Abstract: *Sophisticated attacks usually involve decision logic that observes the victim's responses before deciding the next action. Such logic presents an opportunity for the defence, as it provides a controllable feedback channel. Manoeuvres that manipulate responses can confuse the adversary's decision process, causing them to undertake ineffective actions. By deliberately introducing false information through deceptive manoeuvres, would-be victims can steer adversaries away from their main objectives. In this article, the authors first introduce and analyse a specific deceptive manoeuvre to determine when, where, and how it may be appropriate and effective; and then explore this form of defensive deception from a broader information-warfare perspective.*

Keywords: *Deception, Distributed Denial of Service (DDoS), Security, Decision Logic*

Introduction

Denial of Service attacks (DoS) and their distributed variants (Distributed DoS or DDoS) are one of the earliest attacks devised against computer systems. These attacks exhaust computing resources so that legitimate users are denied access to the system or the services it provides. The most common DDoS attacks are volumetric, during which the adversary causes large volumes of data to be sent to the victim. While effective, these attacks are also highly visible and can be countered by provisioning more resources. However, as recent incidents have shown (Seaman 2017), even in the age of cloud computing and on-demand elastic resource provisioning, an adversary with globally distributed launching pads (for example, botnets) has a distinct advantage. To make matters worse, newer breeds of DDoS attacks have started to appear that are not volumetric, but instead use small numbers of targeted messages to cause resource exhaustion. These attacks exploit flaws in a system's resource-management mechanisms, and in protocol logic and implementation. Unlike volumetric attacks, by the time these low and slow attacks become visible, service has already been denied. Existing mitigations are insufficient, and the attacker's payoff-to-cost (that is, expended-resources) ratio is very high, which makes these attacks a significant threat.

One key resource-management component that is ubiquitous in distributed computing systems is the network stack. Each application interacting with peers over a network depends on a local instance of this stack. In addition, network devices such as routers, switches, and firewalls also incorporate a subset of the layers of the network stack, with ever increasing lev-

els of upper-layer functionality built into them. It is not uncommon to find a full network stack covering all layers in sophisticated routers, application gateways, and firewalls. One disadvantage of handling the various layers of network protocols in tall vertical stacks is that flaws/misconfiguration/mishandling at one layer can disrupt the entire stack, which limits the opportunity to detect and respond to sophisticated attacks.

The research hypothesis that sets the context for the work reported in this article is this: it is possible to defend a services' enclave (that is, a network enclave hosting services provided to clients outside of the enclave) against sophisticated low and slow DDoS attacks by using network manoeuvres that disrupt the adversary's decision logic. By 'network manoeuvring', the authors mean adapting how the network behaves at different protocol layers, both reactively—in response to a suspicious event or observed stress—as well as proactively—on its own. Reactive and proactive manoeuvres can meddle with the adversary's decision logic by presenting fake or misleading responses. These deceptive manoeuvres must be managed carefully to ensure that legitimate clients are not overly impacted. After a brief introduction of the basic concepts of this work, this paper offers a deep dive into the functionality and experimental evaluation of one specific manoeuvre that the authors refer to as SYN Drop, during which the defence chooses to drop subsets of TCP SYN packets (reactively or proactively).

> As demonstrated on numerous occasions, infiltration via social engineering is not uncommon, even in defence systems or by defence contractors. With such a foothold, adversaries can wait and time their attacks during critical phases of a campaign. The authors propose a system that would allow defenders to beat adversaries at their own game by giving them the impression that they are winning and, thus, leading them astray. Cyber defences tend to focus on continued service availability or immunity against future attacks. The authors argue 1) that ignoring the information operation aspect of defensive strategies and tactics is a lost opportunity, and 2) that the directions that can be taken to identify and enhance the *information impact* of the defensive manoeuvres on an *adversary* is a useful complement to, and perhaps a force multiplier for maintaining continuity of service availability under attack. (Wilson 2007)

The use of deception, even purely defensive deception, must be guided by more than measurement and analysis of manoeuvre efficacy against the adversary. This concept of careful and intentional application of deceit, including the consideration of collateral damage, has long been present in traditional military deception, and it is equally relevant as deception is applied in the cyber domain. Deception has the potential to impact both legitimate and malicious users equally; and, in the anonymity-rich world of the Internet, it is often very hard to distinguish between the two. Appropriate awareness and policy must, therefore, govern how deception is used in order to reduce the risk of unintended consequences if the deception is consumed and acted upon by one's own forces or other domestic audiences. The risk applies not only to friendly actors outside of the organisation that is implementing the deception, but also to the administrators, systems, and users of that organisation itself. System administrators must be able to determine what is real and what is fake in their own environments in order to properly administer them, yet the indicators and mechanisms that allow them to pierce the veil of the deception must not be exposed to adversarial actors. Even in the case in which one could ensure that deception is being applied only to the intended targets, all a deceiver can control is the signal received by the adversary, not the impact that signal has on

the adversary's decision processes. Thus, further challenges exist in employing deception in a manner that minimises the space of probable adversary counter-actions to a set that is deemed desirable or at least acceptable.

The two main technical contributions of this article are 1) establishing the feasibility of using the SYN Drop manoeuvre as a mechanism to insert controllable delay into network behaviour without causing unacceptable harm to legitimate clients or meaningful cost to the defender, and in identifying the cases where this manoeuvre will have a positive impact on intelligent DDoS attacks, and 2) analysing the class of deceptive manoeuvres that SYN Drop is part of from an information-warfare policy perspective. The remainder of the article is organised as follows: first a brief overview of the technology suite that enables DDoS-countering network manoeuvres is presented followed by an in-depth description of the SYN Drop manoeuvre and its experimental evaluation in the subsequent section. Next, the questions, concerns, risks, and challenges that arise from the use of deceptive manoeuvres in information warfare are examined from a policy perspective. Afterwards, related work is described, and, finally, the article is summarised and conclusions drawn.

ARMED Overview

The Adaptive Resource Management Enabling Deception (ARMED) (Pal 2017) technology the authors are developing is realised by in-network interception and processing points referred to as the ARMED Network Actors (ANAs). The interception is transparent—the presence of the ANAs is not visible to the parties whose traffic is being intercepted. The processing is protocol-specific and involves monitoring protocol execution, analysis of protocol-specific data for anomalous behaviours, and engaging adaptive manoeuvres. Protocol-specific processing within ANAs makes it possible to split the network stack in a novel way so that processing of two different network-layer protocols, such as TCP and HTTP, can be put on two different ANAs in a tiered manner (see **Figure 1**, below). Since ANAs are transparent, the client experiences its TCP connection and HTTP interactions, for example, as if they were interacting directly with the actual endpoint. However, the client's TCP handshake is performed with the TCP ANA; the HTTP session is established with the HTTP ANA; and actual application-level data and processing take place at the real endpoint. In addition to establishing a deceptive reality, this approach (with redundant ANAs) allows for intelligent ARMED-controlled routing of data through the network to disperse load, consolidate malicious traffic, enable consistency across deceptions, and/or introduce randomness that complicates reconnaissance, and disrupts adversaries that depend on stable and static network paths.

ANA responsibilities

Each ANA is responsible for performing a number of functions organised into layers as depicted on the right side of **Figure 2**, below. At the bottommost layer, an ANA must transparently intercept data that was not originally intended for it. The second layer collects features of interest (for example, HTTP header counts and values, inter-packet delays, current TCP state) from the received packets/flows for analysis by the subsequent profiling and analysis layers that may help in detecting and characterising an attack. The analysis layer itself can consist of multiple analysis plug-ins (for example, a clustering-based anomaly detector, Robust Principal Component Analysis [RPCA] anomaly detection). The Course of Action (CoA) and actuation layers are responsible for determining which defensive manoeuvres to engage and how to parameterise and target them, as well as actually executing them. This is often trig-

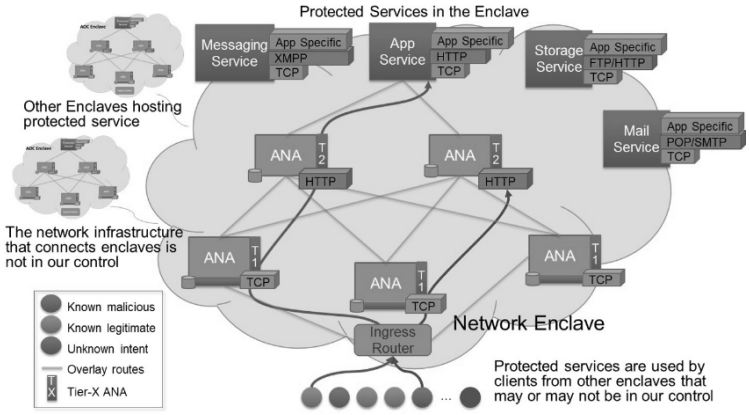

Figure 1: ANA split stack overlay

gered by signals from the analysis layer, but can be triggered by time, network conditions, or other factors—particularly in the case of more proactive manoeuvres.

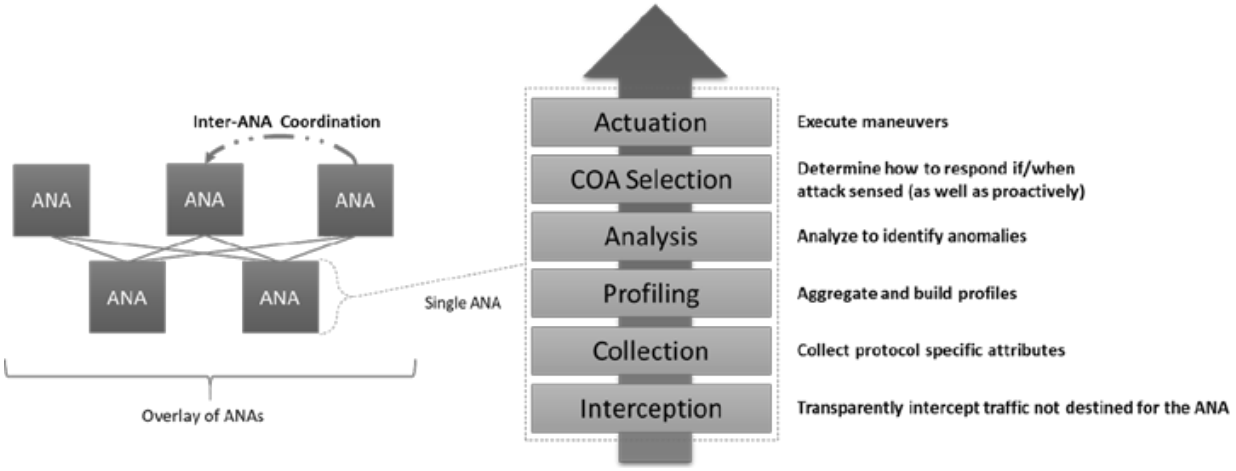

Figure 2: ANA functional areas in context of the overlay

Network manoeuvre examples

The manoeuvres executed by the actuation layer are used by ARMED to disrupt reconnaissance and attack execution. The remainder of this paper will concentrate on a particular manoeuvre, SYN packet dropping, but **Table 1**, below, lists a few others to provide a broader sense of ARMED manoeuvres.

ANAs can also be used to redirect traffic to honeypots and tar pits, or masquerade the true servers/protocols they use.

SYN Dropping

The SYN Drop manoeuvre allows a controlled disruption of the establishment of incoming TCP connections. Such a manoeuvre can be used to inject artificial latency (for example, to give an adversary the false sense that his or her attack is successfully disrupting legitimate service), or to reduce adversary traffic volumes without denying their connections—a useful tactic

Name	Protocol(s)	Description
Fake Hosts	ICMP	Respond to ICMP echo (aka ping) requests for hosts that do not exist
Fake Ports	TCP	Allow establishment of connections to ports that are not actually open
HTTP Redirect	HTTP	Inject a JavaScript redirect into HTML responses to ensure clients have web browser-like capabilities
False Resolution	DNS	Resolve a domain name to an alternate address (for example, to redirect subsets of clients to honeypots, or even to attack themselves)
False Latency	ICMP, TCP	Introduce false latency into responses to some or all clients (to give the appearance, for example, that an attack is working when in fact it is being successfully mitigated)

Table 1: Example manoeuvres

that keeps hidden the defence's awareness of the attack or attack source. The TCP protocol undergoes a handshake during connection establishment. The initial steps of this handshake involve a client sending a SYN packet, followed by the server responding with a SYN-ACK, which acknowledges receipt of the client's SYN. When a SYN packet is lost or dropped, the initiating client will not receive a SYN-ACK and, thus, after a timeout period will resend the SYN. Dropping SYN packets at the ANA is one way to inject artificial latency *without maintaining state on the server.* Alternate ways to inject such latency would require the defence to keep a client's request or response in memory or on disk for a period of time, a potentially costly, memory-intensive operation. It can drop packets for specific IP addresses or universally for all incoming requests. If a source IP is already under suspicion, then this manoeuvre can be narrowly targeted against that IP in a reactive manner. Traditional responses, such as blocking source IP addresses, are effective in limiting access for the given client but have the side effect of sending a signal to that client that their efforts are being mitigated. The advantage with the SYN Drop manoeuvre, particularly in the context of cyber deception, is that it can instead send a signal to the adversary that its attack is working and, thus, response time has been impacted. The SYN Drop manoeuvre can also be engaged proactively (that is, not a targeted reaction to a detection of malice). The experiments described in this paper use SYN Drop proactively, where incoming SYN packets are dropped probabilistically.

The authors performed a set of experiments with this manoeuvre to understand how SYN dropping would impact legitimate and adversarial clients. First, a model of the expected impact on clients was derived as a way to extrapolate to other contexts and configurations. This model estimates the expected additional delay d incurred by a client when ARMED is executing SYN Drop at a given rate—if under normal conditions the connection set-up time is c, then with SYN Drop, the connection set up is expected to take $c + d$.

The model captures how SYN Drop interacts with the clients' TCP back-off parameter σ. A SYN Drop manoeuvre will force the client to retransmit the SYN according to an exponential

back-off strategy, waiting for σ before the first retransmit, and then 2σ, 4σ, ... and so on for the subsequent retransmits before stopping after t failed retransmits. The ordered set of additional delays in connection setup for up to t retransmissions were defined as x, as follows:

$$x = \left\{ \sum_{j=1}^{j=i} 2^{j-1}\sigma \ : \ i \in [1,\, t] \right\}$$

Note that the first delay in the above ordered set is experienced if a SYN packet is dropped; the second delay from the set is experienced if the second consecutive SYN packet sent (after the initial delay) is also dropped. Since this SYN Drop manoeuvre drops SYN requests probabilistically, not all SYN requests will be dropped. X is used as the random variable that denotes the additional delay experienced by a client when establishing a connection. The expected value of X for a SYN Drop with probability p was calculated where x_i represents the $i-$th element in the set, x, as follows:

$$E\left[X\right] = \sum_{i=1}^{i=t} p^i x_i$$

In other words, the expected value of X for a SYN Drop with probability $p$means that the client's expected connection set-up time will be c+ E[X], instead of c.

Note that in proactive SYN Drop, the expected delay is the same for benign and malicious clients. Its efficacy thus depends on the differences in the behaviour of good or bad clients. For instance, if a typical malicious connection remains active for 200ms and a typical benign connection remains active for a longer period, such as 2s, then a SYN Drop causing an expected 200ms delay will nearly double the time required to complete the adversarial task, but only slow the benign client by about 10% (as only connection establishment is affected by this particular manoeuvre). This advantage becomes even more potent if the malicious requests are executed at a high rate. Experiments to validate this perspective, described in more detail below, fell under two broad categories: 1) benign and malicious clients had similar interactions (for example, similar duration, similar requests) and 2) legitimate and malicious clients had different interactions patterns.

Probability	**Delays for Given Back Off (ms)**				
	250	**500**	**1000**	**1500**	**2000**
$p = 0.05$	14.59	29.19	58.38	87.56	116.75
$p = 0.10$	34.25	68.50	137.00	205.50	274.00
$p = 0.15$	60.28	120.56	241.13	361.69	482.25
$p = 0.20$	94.00	188.00	376.00	564.00	752.00
$p = 0.25$	136.72	273.44	546.88	820.31	1093.75
$p = 0.30$	189.75	379.50	759.00	1138.50	1518.00
$p = 0.35$	254.41	508.81	1017.63	1526.44	2035.25
$p = 0.40$	332.00	664.00	1328.00	1992.00	2656.00

Table 2: Performance seen by clients when system is under attack relative to the SYN Drop probability

Table 2, above, depicts a client's expected incurred delay (in milliseconds) from a SYN Drop for probabilities 0.05 to 0.40 using 0.05 size intervals across a number of potential back-off periods. The back-off period used by the clients in the test environment was experimentally determined to be 1000ms. Clients were configured to declare that the server was down for that connection attempt if it exceeded five seconds. With a back-off period of 1000ms, t needs to be at least three to make x_t equal or larger than the 5s timeout period, making $x = \{1000, 3000, 7000\}$. As higher probability values for the SYN Drop, the expected delay increases. The actual experienced delay increase may be slightly different due to calculating the expected value with 7000ms, while the clients were configured to wait only 5000ms before declaring the server as down.

For all experiments, the testbed employed, shown in **Figure 3**, below, was realised in an OpenStack-based private cloud. All nodes ran Ubuntu 16.04 with one virtual processor and 1GB of RAM, with the exception of the ANA nodes, which ran Fedora 24 with two virtual processors and 2GB of RAM each.

Figure 3, below, compares the observed results in the environment with the predicted values calculated above. In this test, each client makes 2 HTTP GET requests per second for files of three sizes: 100K, 50K, and 10K. The time it takes to complete each request is averaged over the five minute test time. To calculate the average delay, the average connection set-up time for the baseline (no SYN Drop, no attack) is subtracted.

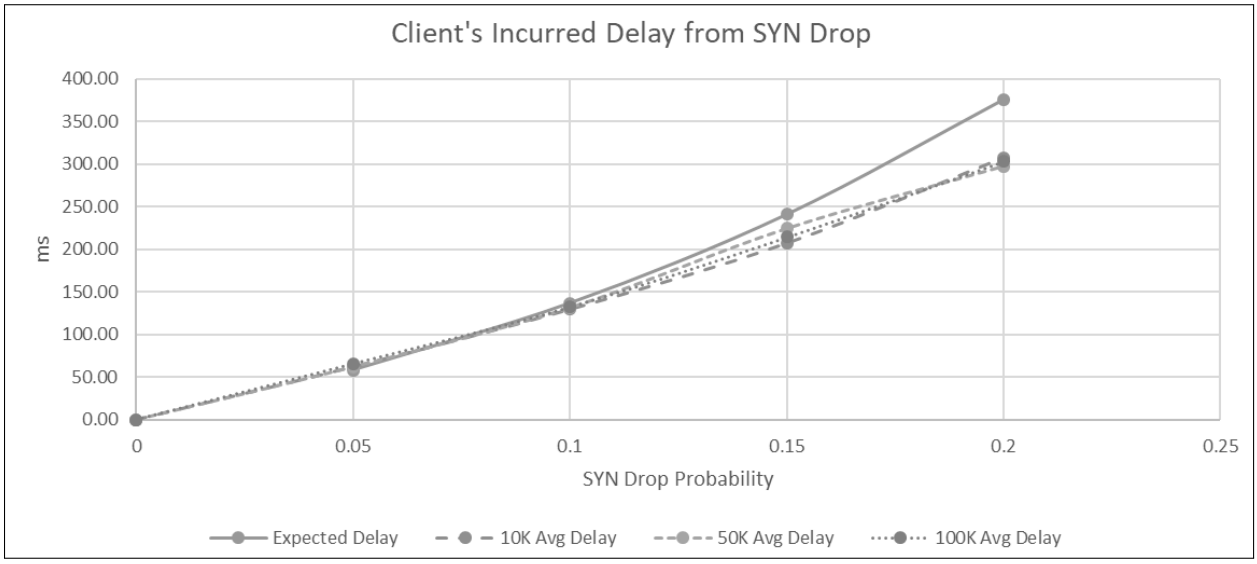

Figure 3: Observed vs expected average delays with the SYN Drop manoeuvre

As expected, the actual incurred delays are slightly less than the predicted values (as mentioned above, due to the 5s timeout). This difference increases as the SYN Drop probability increases. Otherwise, the experimental results closely track the results predicted by the theoretical model.

Reactive SYN Drop manoeuvre

In a reactive or targeted use of SYN Drop, a detection/suspicion event leads the system to drop the SYN packets from suspected IP addresses. In this case, both the overhead for ARMED and the collateral damage on legitimate clients are nearly non-existent. Since the SYN Drop

manoeuvre puts the onus on the client to retain a copy of the packet, the only state ARMED needs to maintain is the list of clients for which SYN dropping should occur. Since only packets from suspected clients are dropped, legitimate users see no additional latency. The efficacy of such a manoeuvre is thus a function of 1) the accuracy of the detection or suspicion, and 2) the extent to which the manoeuvre disrupts the adversary's logic. The former, while in the scope of the ARMED technology, is not the focus of this paper. The latter leads to complex questions about the particular adversary. (For example, is a set of attack code or a human attacker being deceived?) What can be more readily measured in this context, however, is if the manoeuvre slows down the targeted IP sources (as captured above) and, as a result, reduces the impact of an ongoing attack. Experiments with reactive SYN Drop show that the targeted clients are, in fact, slowed down and their ability to harm legitimate clients is severely reduced.

Proactive SYN Drop manoeuvre

The effectiveness of a proactive SYN Drop is far less clear cut. Without suspicion/detection to discriminate between good and bad, the impact on legitimate clients is potentially much greater. In these experiments, an evaluation tool, developed by SRI International called DDoS Laboratory, was leveraged. The tool uses a feedback loop to adapt its attacks to the perceived performance (of the victim server) to achieve a target latency degradation. The first attack, referred to here as 'Max Clients', continually creates new connections to the server to retrieve a file in an attempt to exhaust the server's limit of allowed connections. It continues to increase the number of connections until it reaches a target average response time for the server. The target delay in these experiments is five seconds. The second attack, referred to here as 'Max CPU', makes requests that trigger compression of the response in an attempt to exhaust available server CPU.

When the Max Client attack is used against the system with no manoeuvres activated, performance degradation to the target level occurs. In the first experiments, the attacks for 300 seconds with a target response time of five seconds were tested, with good clients making requests to files of different sizes: 10K, 50K, and 100K. 100K is also the size of the file the attack tool is using to probe the server to adapt the attack load. Clients requesting a file of the same size are, therefore, expected to observe the target level of degradation that the attack is imposing on the server.

The three pairs of graphs in Figure 2, below, show the delay experienced by these clients over the course of three different experiments (bottom row of graphs) and the client-observed availability (in the top row of graphs, light grey indicates available, and darker vertical bands unavailable). The columns, from left to right, are for files of size 100k, 50k, and 10k. In the availability/uptime graphs, the Y axis represents the number of clients that perceived the server to be available, and the X axis represents time. In the first response time graph (bottom row), the Round Trip Time (RTT) increases until it averages five seconds. In the two tests with the smaller graphs, the clients' RTTs do not increase as much, but they still see a noticeable effect

With this set of baseline results, the experiments were repeated, but with the SYN Drop manoeuvre enabled in a *global proactive* manner. Due to the fact that the legitimate clients and malicious clients employed similar interaction patterns (for example, both requested files of similar sizes on a periodic basis) in these experiments, and the fact that SYNs were dropped

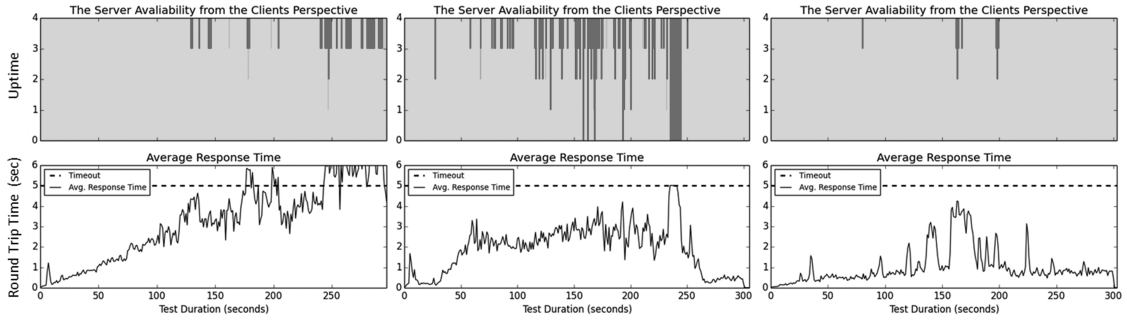

Figure 4: Client-perceived server performance requesting files while a Max Client attack was underway; the first graph shows clients requesting a 100K files, the second a 50K file, and the third a 10K file

with equal probability across all clients, the results showed no reliable benefits for the defence; in some experiments a slight benefit was seen, and in others a slight detriment.

Since the SYN Drop manoeuvre delays connection establishment, the authors hypothesised that it will be more significant in situation where legitimate clients make longer-lived connections than the malicious ones.

To test this hypothesis, the authors constructed clients that make a number of GET requests *over the same long-lived TCP connection* to an Apache v2.4.7 web server. In these tests, clients send 22 GET requests to the server with a 0.25 second delay between each request. First, the impact of SYN Drop without an attack was tested. Next, the Max Client attack was used against the server and the effectiveness of the SYN Drop manoeuvre was inspected, as seen by the client using the service, with and without the manoeuvre.

In **Table 1**, above, under attack with no defence, the clients see greatly reduced performance. On the left, when the system was not under attack, clients took an average time of 5.62 seconds (bottom graph) to perform their tasks, and the clients did not experience any unavailability (top graph). However, on the right (system under attack), clients took an average time of 19.45 seconds to perform the same tasks (bottom graph). After approximately 300 seconds, the clients started reaching their timeout (30 seconds), which caused some of them to declare the service as 'down' (as shown by the darker bands in the top graph).

The authors experimentally derived an effective SYN Drop probability for this attack type, varying probability and target delay values as depicted in **Table 2**, below. Using the identified 55% probability, client performance was greatly improved with the SYN Drop manoeuvre, compared with an unprotected system.

Experiments have shown the potential for a very effective deceptive use of SYN Drop in a reactive/targeted manner, as well as, in some contexts, a proactive/untargeted manner. There are, however, possible ways that an adversary could game a SYN Drop. For example, since the delay incurred by the client is based on the SYN Drop probability and the client's back-off period, an uncommonly small back-off value used by the attacker would cause the manoeuvre to have a disproportionate impact against *legitimate* clients. However, in order to do this, the adversary must recognise that a SYN Drop is being used and perform the analysis to determine an effective back-off value.

Policy Questions, Risks, and Challenges with Deceptive Manoeuvres

The technology described, although important in the context of information warfare, is not an information warfighting tool by itself. It is primarily a DDoS defence mechanism, which has an information impact on the adversary. However, defence mechanisms designed to impact the adversary in the information domain raise an interesting opportunity. If the adversary is engaged in information warfare (for example, against social media services) or hosts in the defended enclave contain bots that are engaged in an information-warfare campaign, is it possible to use deceptive manoeuvres such as SYN DROP as a counter measure; and, under what conditions is this useful, safe, and advisable?

The authors present three main areas of consideration that are pertinent in this context:

- Collateral damage: Proactive use of SYN DROP delays benign and legitimate users alike. This impact can be minimised by targeting the manoeuvre, by applying heuristics (such as employing the manoeuvre only when current readings project near term resource exhaustion), or by using only in scenarios where the legitimate users hold long-lived connections (minimising the impact on them). Even with the above techniques, some impact may still be present. Collateral damage, therefore, must be a consideration before deploying a deceptive manoeuvre in cyber-defence as well as in information warfare. Proactively dropping connections to social media servers hosted in the defended enclave or slowing client sessions may impact the activity of the adversaries trying to form or foment an opinion on the social media platform, but it may also slow down the posts and sharing of normal users. It is worth studying how delays impact viral spreading, and the resulting impact on information warfare campaigns.

- Added complexity: Traditional deception in the military domain requires careful advanced planning and incurs additional capital and operational expenses. Deception in the cyber domain is no exception. While this article primarily considered defensive deception, in the context of information warfare, deception is often used for offensive purposes (for example, influencing news, ideas, and opinions of adversary populations or decision makers). Regardless, assets, access, and capabilities must be planned and be in place before a deception campaign can be mounted. Cost factors aside, an extra layer of complexity arises from having to maintain two *separate books* to manage the infrastructure and perception. System administrators must be able to understand what is truly going on in their networks, to see reality and not the deception.

- Ethical concerns: While collateral damage is often examined purely in terms of the tangible impact on legitimate users, the ethical and legal issues are less tangible. Is it acceptable to slow down competitor traffic/content or traffic from users who pay less than others? Ethical guidelines and legal frameworks exist to navigate these issues. Similarly, deceptive manoeuvres used to cause a material impact on unsuspecting benign users may also fall under the scrutiny of existing ethical guidelines and legal frameworks.

Related Work

Traffic scrubbing, elastic provisioning and scaling, distribution and dispersion of resources (for example, content distribution networks), and use of alternative routing are common detec-

tion and mitigation mechanisms for volumetric DDoS attacks. While effective against volume-based attacks, these solutions are typically not equipped to mitigate many low and slow attacks in which legitimate and malicious traffic are harder to disambiguate. Next Generation Firewalls incorporate inspection of upper-layer protocols, but still act as signature-based all-or-nothing packet filters and cannot easily implement manoeuvres such as SYN Drop without disclosing the filtering to the attacker. The techniques being developed in ARMED focus on low and slow attacks and can work alongside traditional mechanisms to counter both volumetric and non-volumetric attacks.

Proactive defences that disrupt reconnaissance and attack execution by injecting randomness into network routing have been explored (Keromytis, Misra & Rubenstein 2004; Shan, Neamtiu, Qian & Torrieri 2015; Lu, Marvel & Wang 2015). Often the proactive nature of the defence comes with either restrictive assumptions (for example, a fixed known set of legitimate clients), or too high a cost to be generally deployed. These are the questions that this paper attempts to address for a particular manoeuvre: SYN dropping. In general, ARMED takes a mixed proactive/reactive approach and, in both cases, explicitly models the impact of a manoeuvre on legitimate clients, and uses this information along with the severity of the threat to select defensive manoeuvres.

Other work (Al-Duwairi & Manimaran 2005; Sun *et al.* 2007; Huitema, Sanders & Kaniyar 2008) has explored SYN dropping from the perspective of mitigating SYN floods, a previously popular form of DDoS attack that has largely been addressed in recent years. In contrast, the authors are exploring SYN dropping as a mechanism to introduce deceptive latency with little cost to the defender to counter various forms of DDoS attack beyond SYN floods.

There has been work in building manoeuvre frameworks (Beraud *et al.* 2011; Soule *et al.* 2016) that overlap with some of the core goals of ARMED, but have been applied either to the Moving Target Defence (MTD) space or have incorporated deception, but are focused on host-based manoeuvres.

Effectively measuring security is a well-known R&D challenge, and a wide body of work already exists. Early work in measuring moving target defences and deception has begun (Moody, Hu & Apon 2014; Atighetchi *et al.* 2016; Soule *et al.* 2015). The authors plan to build upon these concepts in evaluating their own deceptive network manoeuvres.

Work exists in understanding the risks and resulting policy considerations of both traditional (Daniel & Herbig 2013) and cyber (Heckman *et al.* 2015; Wilson 2007) deception. This article explores these aspects from the perspective of defensive cyber deception and, specifically, against the backdrop of network manoeuvres.

Conclusion

This article presents a specific case of modulating *normal* network features or behaviours for cyber defence. Adaptive manoeuvres (such as packet dropping or delaying connection establishment) have the inherent advantage of minimising harm to legitimate users (since applications are built to tolerate some amount of these behaviours) and are part of a larger exploration of deception in cyber defence. As this paper demonstrates, simple proactive application of these manoeuvres (applied indiscriminately) may only be effective in specific contexts—but when *suspicion* or observed *stress* is used, the result is very effective at manipulating adversary

decision processes. For example, since the adaptive attacks described above rely on signals from the system to determine current impact, the SYN Drop manoeuvre is able to manipulate those signals to deceive the adaptive attack logic. When other manoeuvres (such as those in **Table 1**, above) indict non-compliant clients, targeted SYN Drop has an even stronger impact on the attack logic.

This article focuses on one manoeuvre within one protocol (though one applicable to the majority of typical network traffic). The authors are still at an early stage in their exploration, and there are many protocols and many manoeuvres to evaluate. Evaluating the effectiveness of security technology is hard, and evaluating the effectiveness of deceptive cyber-defence is no exception. More work is clearly needed. The authors plan to explore additional protocols and manoeuvres, as well as alternate ways to evaluate the deceptive manoeuvres, including new attacks and attack effect injection mechanisms, as well as quantitative analysis of the impact of deceptive manoeuvres on an adversary's logic.

In the case of deceptive manoeuvres that inject false information into the data stream, determining a defence's efficacy is only part of the necessary process to determine its appropriateness. The potential for causing negative effects (from the defender's perspective) either by misleading legitimate users or by misleading attackers into taking unexpected and undesirable actions means that a deeper analysis is important to understand the full impact of deploying such defences. This work lays out an initial set of risks, challenges, and concepts that must be considered from a policy perspective when working with deceptive network manoeuvres.

Acknowledgements

This work was funded under DARPA contract HR0011-16-C-0058. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

Al-Duwairi, B & Manimaran, G 2005, 'Intentional dropping: A novel scheme for syn flooding mitigation', *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005*, IEEE, vol. 4, pp. 2890–4.

Atighetchi, M, Benyo, B, Eskridge, TC & Last, D 2016, 'A decision engine for configuration of proactive defenses—challenges and concepts', *Resilience Week (RWS)*, pp. 8–12.

Beraud, P, Cruz, A, Hassell, S & Meadows, S 2011, 'Using cyber maneuver to improve network resiliency', *Proceedings of the 2011 Military Communications Conference, 2011-MILCOM*, IEEE, pp. 1121–6.

Daniel, DC & Herbig, KL, eds., 2013, *Strategic military deception: Pergamon policy studies on security affairs*, Pergamon Press, New York, NY, US.

Heckman, KE, Stech, FJ, Thomas, RK, B, Schmoker & Tsow, AW 2015, *Cyber denial, deception and counter deception: A framework for supporting active cyber defense*, Springer, CH.

Huitema, C, Sanders, HL & Kaniyar, SN 2008, *System and method for defeating syn attacks*, US Patent 07391725.

Keromytis, AD, Misra, V & Rubenstein, D 2004, 'SOS: An architecture for mitigating DDoS attacks', *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 176–88.

Lu, Z, Marvel, L & Wang, C 2015, 'To be proactive or not: A framework to model cyber maneuvers for critical path protection in MANETs', *Proceedings of the Second ACM Workshop on Moving Target Defense*, ACM, pp. 85–93.

Moody, WC, Hu, H & Apon, A 2014, 'Defensive maneuver cyber platform modeling with stochastic petri nets', *Proceedings of the 2014 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, IEEE, pp. 531–8.

Shan, Z, Neamtiu, I, Qian, Z & Torrieri, D 2015, 'Proactive restart as cyber maneuver for android', *Proceedings of the Military Communications Conference, MILCOM 2015*, IEEE, October, pp. 19–24.

Soule, N, Pal, P, Clark, S, Krisler, B & Macera, A 2016, 'Enabling defensive deception in distributed system environments', *2016 Resilience Week (RWS)*, 16-18 August, pp. 73–6.

Soule, N, Simidchieva, B, Yaman, F, Watro, R, Loyall, J, Atighetchi, M, Carvalho, M, Last, D, Myers, D & Flatley, B 2015, 'Quantifying & minimizing attack surfaces containing moving target defenses', *2015 Resilience Week (RWS)*, pp. 1–6.

Sun, C, Fan, J, Shi, L & Liu, B 2007, *A novel router-based scheme to mitigate SYN flooding DDoS attacks*. Student poster, *IEEE INFOCOM 2007*.

Wilson, C 2007, *Information operations, electronic warfare, and cyberwar: Capabilities and related policy issues*, Library of Congress, Congressional Research Service, Washington, DC, US.