17th International Command & Control Research & Technology Symposium

"Operationalizing C2 Agility"

Adopting Botnet Herders' Techniques in Military C2 Systems

Paper number I-031

Topic
1. Concepts, Theory, and Policy

Name of Author
Tim Grant

Professor, Operational ICT, Faculty of Military Sciences

Point of Contact
Tim Grant

Netherlands Defence Academy, PO Box 90.002, 4800 PA Breda, The Netherlands

Tel: +31 (0)638 193 749

Till 30 June 2012: tj.grant@nlda.nl From 1 July 2012: tim.grant.work@gmail.com

Adopting Botnet Herders' Techniques in Military C2 Systems

Abstract:

A botnet is a network of hundreds, thousands or even millions of compromised computers. The botnet is monitored and controlled automatically by Command & Control (C&C) servers, with the human "botnet herder" exercising overall supervisory control. Over the past two decades, a variety of malware and communication technologies and botnet topologies have been developed. Many of these are an advance on those seen in present-day military Command & Control (C2) systems.

The purpose of this paper is to identify the techniques that botnet herders have developed to see whether they could be adopted in military C2 systems and, if so, what operational benefits they could bring. The operational environments, threats, responses, organizational structures, and software architectures of military C2 systems and of botnets are summarized. These features are then compared, leading to the recommendation that botnet herders' techniques for stealth and obfuscation could be adopted to replace the current practice of air-gapping.

Tim Grant

Professor, Operational ICT, Faculty of Military Sciences

Netherlands Defence Academy, PO Box 90.002, 4800 PA Breda, The Netherlands

Tel: +31 (0)638 193 749

Till 30 June 2012: tj.grant@nlda.nl From 1 July 2012: tim.grant.work@gmail.com

Introduction

Harris and White (1987, p.1) summed up the state of C2 research 25 years ago as: "There is no theory of C2; there is an informational chaos; there is a technological revolution; [and] there is an organisational chaos." Since then, we have made progress in the first of these, thanks to developments in Network Enabled Capabilities (NEC). In NEC theory, we might think of the NEC tenets or value chain, the spectrum of C2 approaches from "Power to the Edge", the doctrine, the factors of organization, training & education, materiel, leadership, personnel, facilities, and interoperability (DOTMLPFI), the four domains (physical, information, cognitive, and social), and the NEC Maturity Levels (NMLs). In the underlying scientific disciplines we might think of developments in knowledge in the areas of human supervisory control, situation awareness, software architectures, the emergence of Commercial Off-The-Shelf (COTS) software, the Internet, the web, and social media.

Unfortunately, our (potential) opponents have also been busy. In this paper we will be concerned with cyber-opponents, and in particular those who deploy and employ botnets. While botnets can have benevolent purposes (e.g., SETI@Home), we focus on malevolent botnets because these have

necessarily evolved in an adversarial environment. A botnet is a network of compromised computers ("zombies") infected with one or more malicious software agents ("bots"), usually connected through the Internet (Puri, 2003) (Li, et al, 2009). The person who controls a botnet is known as a "botnet herder" (or "botmaster"). The infrastructure used by the botnet herder to monitor and control his[1] botnet is known as "Command & Control" (abbreviated "C&C"). Botnet C&C consists of a communications channel and one or more C&C servers (Zeldanloo & Manaf, 2009). The C&C servers may themselves be zombie-hosted bots. They relay commands from the botnet herder to the bots and, depending on how the botnet is used, they may also relay data collected by the bots back to the herder. Traditionally, the communications channel is implemented using Internet Relay Chat (IRC) technology, although Hypertext Transfer Protocol (HTTP) can also be employed. More recently, botnets have begun to use social media (e.g., Twitter) as their C&C channel (Kartaltepe, et al, 2010). The full range of C&C topologies exist, from centralised hierarchies to fully decentralised peer-to-peer networks. The topology may include redundant C&C servers, providing resilience against server failure or shutdown. Depending on the herder's business model, C&C redundancy also allows the herder to lease out parts of his botnet.

Botnet herders take pains to make their botnets and the C&C traffic difficult to detect. The reason is that they need to protect their assets not only from legal shutdown but also from hijacking by other botnet herders. They have developed a variety of techniques, including programming bots to "call home" to alternative servers at random or pre-scheduled times if C&C communication is lost. Another technique includes redirecting Internet Protocol (IP) and domain-name lookup requests, C&C traffic, and the extraction of collected data through blind proxies. A particularly interesting technique, somewhat akin to frequency hopping, is "fluxing", in which the IP address and/or domain names of each bot and C&C server are constantly changing.

The research question addressed in this paper is whether any of the techniques that botnet herders have evolved could be usefully adopted in military C2 systems. Botnets are extremely successful, with a study in 2008 showing that, on a typical day, about 40% of the 800 million computers connected to the Internet is a bot. In many ways, they seem to employ operationally many of the capabilities that the NEC community see as future goals. For example, botnets already routinely exhibit swarming, edge topology, and the need-to-share principle. One could say that botnets are already at NML 4. Over and above this, they also incorporate certain techniques that the military C2 community has not yet even thought about, like fluxing. These techniques enable botnets to survive undetected while being completely immersed in the "dirty" Internet without many of the defensive security measures that military C2 system designers see as essential, such as "air-gapping".

The purpose of this paper is evaluate the techniques that botnet herders have developed to see whether they could be adopted in military C2 systems and, if so, what operational benefits they would bring. The paper consists of five sections. After the introductory section, the relevant features of military C2 systems will be summarised in the second section. The third section summarises the equivalent features of botnet C2 technology. In the fourth section the botnet techniques and their military C2 equivalents will be compared, and the potential operational benefits to military C2

[1] Botnet herders are almost exclusively male.

systems from applying suitable botnet techniques will be identified. Finally, conclusions will be drawn and recommendations made in the fifth section.

Relevant Features of Military C2 Systems

The relevant features of military C2 systems are discussed in terms of the operational environment, threats and responses, organizational structure, and C2 system architectures. Since the latter two features are under transformation to NEC, they will be discussed twice: once for Industrial Age C2 and a second time for Information Age C2.

Operational environment

The operational environment in which modern complex endeavours involving military forces take place are characterised by the close interaction between actors, activities and events in the physical (geographical) domain, and to a lesser extent in the information, cognitive, and social domains. Organizational structures and legal constraints are considered here as being part of the social domain. The terrain within which the actors manoeuvre is an important determinant on how they can act. Another determinant is their set of technical capabilities, so that a land force that has helicopters can travel faster and further than a force that only has road vehicles.

Threats and responses

Operationally, military forces are threatened by the opposing forces. The nature of the threat varies according to whether the opposing forces are another military, insurgents, or natural forces. Military forces must comply with national and international law regarding their conduct, as well as take into account the prevailing public opinion. Threats specific to C2 systems are that the information that they hold may become known to other actors, that the systems may be destroyed (by kinetic or cyber means) or disrupted (by engineering failures or by enemy action), or that the users may be unable to gain or be denied access to their C2 system. Grant, et al (2007) have enumerated qualitatively the ways in which C2 systems are fallible.

In response to these threats, C2 systems are designed and operated so as to maintain the principles of Confidentiality, Integrity, and Availability (CIA). Confidentiality applies to the information within the C2 system, requiring that this information should not be disclosed to unauthorized individuals, organizations, or systems. This principle is often expressed as “need-to-know”. When applied to the information within a C2 system, the integrity principle requires that it must not be possible to modify the information within the C2 system undetectably. Data integrity is a stricter term meaning that information in the C2 system must faithfully represent the true state of the object that the information represents, where representational faithfulness comprises the four qualities of completeness, correctness/accuracy, timeliness/currency, and validity (Boritz, 2004). Some authors add precision as a fifth quality. The integrity principle also applies to the C2 system’s hardware and software, where it means that the system must constitute a complete working whole. The availability principle means that the C2 system and the information within it must be continue to be available to its users, despite disruptions due to power failure, hardware or software failures, or denial-of-service attacks. Thus, failure or damage could result in the loss of integrity or availability, depending on their effects.

Because military forces may have to account for their actions, e.g., to political decision makers or in a court of law, some authors add the two principles of Authenticity and Non-Repudiation. Authenticity guards against tampering with a message, modifying it, or including extraneous material (e.g., a virus). Most C2 systems are distributed, making it necessary to ensure that the data, documents, transactions, and communications exchanged between parts of the system are authenticated and that the parties to the exchange are who they say they are. If the principle of Authenticity can be violated, then there is a threat that an unauthorized individual, organization or system can masquerade as one with "need-to-know". Non-repudiation means that neither party can deny that they took part in the exchange. Violation of the principle of Non-Repudiation would mean that an individual, organization or system, whether authorized or not, could deny their involvement in a set of events. Both principles are usually achieved by encryption. For simplicity, we will include Authenticity and Non-Repudiation as aspects of the principle of Confidentiality.

Responses to these threats are usually defensive and take a variety of forms depending on which CIA principle is being threatened. Confidentiality is maintained by encrypting information stored in or passing through the C2 system, by physical security (i.e., control both on access to where the system is located and on access to the services it provides), by personal security (e.g., by vetting C2 system users, by checking biometric data or an identity token, or by requiring a password before access to the system is granted), by physical separation of the C2 system from other systems ("air-gapping"), and by controlling electromagnetic emissions from the system. Integrity is maintained by: logging; (data, hardware, and software) redundancy; technological diversity; taking regular backups and providing roll-back facilities; and the like. Data integrity is maintained by applying data structure, derivation, retention, and value rules. Availability is also guaranteed by redundancy and diversity.

Military hierarchy & Industrial Age C2

Organizational structure has been extensively studied. Classic references include Galbraith (1974; 1977; 2008) and Mintzberg (1980). Ideas on how to classify organizations vary from Galbraith's (1974) early focus on information processing to his more recent star model balancing strategy, structure, people, processes, and reward (Galbraith, 2008). By contrast, Mintzberg focused on five basic parts of an organization: the operating core, the strategic apex, the middle management, the technostructure, and the support staff. Coupled with five coordinating mechanisms, nine organizational design parameters, and four sets of contingency factors, Mintzberg identified five structural configurations: Simple Structure, Machine Bureaucracy, Professional Bureaucracy, Divisionalized Form, and Adhocracy. Nissen (2005) shows that Mintzberg's Machine Bureaucracy most closely corresponds to the traditional military organizational hierarchy. Although other approaches play a role, coordination in military C2 centres on the standardization of policies, doctrine, standard operating procedures, task checklists, detailed work instructions, and processes. The nature of military leadership is highly controlling, and military personnel are extensively trained and indoctrinated. Military units are principally grouped by function, and many include a specific function devoted solely to planning. Decision making is very centralized. The corresponding C2 processes and systems have been termed "Industrial Age C2" (Alberts, Garstka & Stein, 1999). Industrial Age C2 system architecture emphasizes the flow of formal communication up and down the organizational hierarchy, with situation reports (SITREPs) flowing from the operating core upwards through the middle management to the strategic apex and Operation Orders flowing

downwards from the strategic apex. In an empirical analysis of eleven C2 systems, Grant, et al (2011) showed that they had a scale-free topology, corresponding to a hierarchical organizational structure.

Edge organization & Information Age C2

Alberts and Hayes (2003) introduced a fresh approach to organizational design – the Edge organizational structure – which appears to be particularly appropriate to military warfare in a modern NEC environment. In contrast to the traditional hierarchical structure, Information Age C2 moves decision making from the strategic apex down to the edge of the operating core. Nissen (2005) shows that it is not possible to identify the Edge organization with any one of the Mintzberg (1980) archetypes. Instead, it reflects elements of the Adhocracy, Simple Structure, and Professional Bureaucracy. Nissen labels the combination Professional Adhocracy.

Instead of being based on standardization, the coordination mechanism in Information Age C2 centres on mutual adjustment (Nissen, 2005). In an Edge organization, warriors will be required to exhibit more generalism and will be given greater autonomy over their tactics and actions. Unit size will be relatively small, enabling their combination dynamically in response to varying environments and missions. Training and indoctrination is likely to continue to be high, as modern warfare requires knowledge-intensive skills. Some functional groups are likely to persist, albeit more focused on achieving objectives rather than producing outputs. Decision making will be selectively decentralized. The corresponding C2 system architecture will be networked. Nissen notes that few examples of Edge organizations can be identified in practice, citing non-military instances in university research, open-source software development, and soccer. In the military domain, special forces teams exhibit many Edge characteristics. Monsuur, et al (2011) provide the mathematical tools for coupling models of C2 networks in the physical, information, cognitive, and socio-organizational domains.

Relevant Features of Botnet C2

In this section, the relevant features of botnet C2 are also discussed in terms of the operational context, threats and responses, organizational structure, and C2 system architectures.

Operational environment: cyberspace

The operational context in which botnets are employed centres on the information domain, and has close interactions with the cognitive domain. Involvement in the social domain is limited, with organizational structures being simple. Botnet herders are almost entirely free from legal constraints and from the physical domain. Their botnet can be spread over the world, and their operations can cross borders and jurisdictions as if these barriers did not exist. The “terrain” within which botnets are embedded – cyberspace – has entirely different characteristics to the physical mountains and valleys that determine how kinetic operations can unfold. Likewise, the technical capabilities of bots are very different to tanks, ships and aircraft.

A botnet consists of a network of hundreds, thousands or even millions of infected computers (Puri, 2003) (Li, et al, 2009). A botnet herder can purchase or hire a botnet or construct one himself. To construct a botnet, the herder must find computers that are vulnerable and then exploit this vulnerability to upload his bot. The vulnerability may be in the computer hardware, its operating system, or its application software or in the communications network linking the computer to

others. More commonly, the botnet herder uses social engineering techniques (Mitnick & Simon, 2002) to persuade the computer's unsuspecting user to upload the bot, e.g., by sending the uploader to the user as an attachment to an email or by attracting the user to a website where the uploader is hidden in the webpages to be sent to the victim's computer. The uploader then uploads the malicious software ("malware") payload from the herder's server and installs and runs it. This payload is the bot. The bot's capabilities depend on the herder's goals. It will invariably have the capabilities of taking over control of the host computer, of hiding its activities ("stealth"), and of "calling home" to the herder's C&C servers. In addition, it will usually have capabilities for monitoring information stored on or passing through the host computer, for sending copies of the monitored information to the C&C servers, for changing the information, and for denying some or all of the host computer's services to the user. Extracting information violates the host's Confidentiality, changing information violates Integrity, and denying services violates Availability. The bot may also have the capabilities of relaying information extracted by another bot, of infecting other computers to which it is linked, or even of acting as a C&C server.

The bot will have an IP address. In cyberspace, an IP address is equivalent to the latitude and longitude for a location in physical space. If the herder has not hardcoded the IP address into the bot, it can obtain its IP address from the Internet's Domain Name Service (DNS). The DNS notes in its routing table the bot's domain name and how messages can be routed to it. The bot can easily (and quickly) change its IP number (and its domain name), either at a preset time, or when a particular event occurs, or when it receives a command to do so from the botnet herder via the C&C servers. A change in IP number is equivalent to moving to another location in physical space in a fraction of a second. Moreover, this can be repeated, equivalent to hopping around the world in a way no physical vehicle could do or to frequency hopping in the electromagnetic spectrum. Changing domain name and/or IP number is known as "fluxing".

Botnets are constructed by two methods. In the first method, the botnet herder uses a software tool to scan ranges of IP numbers automatically to see if the computers at these addresses are vulnerable to exploitation. Vulnerable computers are penetrated, and bots uploaded to them. These bots then join the botnet by "calling home". This is the method favoured by inexperienced hackers, often derogatively known as "script kiddies". In the second method, social engineering is used to infect computers using malicious email or websites (Mitnick & Simon, 2002). The infection is spread using viruses or worms. The second method is favoured by more professional hackers. There is a mature market in vulnerability scanning tools, bots and botnet construction kits, malware, botnets, and the associated services (Kamluk, 2008).

Direct control of the botnet is impractical. The botnet herder's computer would be overloaded if thousands of bots sent information directly to the herder. Moreover, it would be all too easy for defenders of the infected computers to discover the herder's identity by studying the software code of a single captured bot. For these reasons, botnet herders use an intermediate layer of computers – C&C servers – to monitor and control the bots automatically. The C&C servers control the bots directly, and the herder supervises the C&C servers. When setting up their botnets, herders need to make choices about the topology of the C&C servers, about the communication channels from herder to C&C and from C&C to the bots, and about the communication protocols to be used. Commercially-available tools and botnet construction kits can make these choices easy. The choices

available to and their implications for botnet herders are discussed in more detail in the subsequent section on botnet organization structure and C&C.

A botnet can be used operationally for a wide range of malicious purposes (Czosseck & Podins, 2011). Most well known is the Distributed Denial of Service (DDoS) attack, in which the bots are instructed to send queries to a target, tying up the target's resources to answer these queries. In doing so, the target has too few resources to respond to genuine queries from bonafide clients, denying its services to them. DDoS attacks use unsophisticated bots in a brute force manner, do not require the more-difficult penetration of the target, and are simple to set up. Less well known is click fraud, in which bots automatically and repeatedly click on pay-by-the-click Internet advertisements, thus generating income for the herder. However, this requires target penetration. A step-up in sophistication is mass email spamming, in which the bots must be capable of generating a large number of emails, perhaps using an address book belonging to the unsuspecting user of the host zombie and tailoring the emails to look as if they have been humanly authored. These types of attack have in common that the target is external to the botnet.

There are also several types of attack where the zombies are themselves the targets. Since the zombies have already been penetrated to install the bots, this step does not represent a barrier to what the botnet can do. The first two types of attack depend on their non-stealthiness. The first type is website defacement, where the website owner's message is replaced by the botnet herder's message. The herder's motivation is usually political. The second type is scareware, where the user is denied the use of his/her system until some form of ransom is paid. The herder's motivation is usually financial gain. The remaining types of attack are invariably stealthy. Spyware captures details of how the target system is used, usually for marketing purposes. The motivation is financial gain. A step-up is information extraction, in which login identities and passwords, credit-card numbers, application serial numbers, emails, other sensitive information is captured and extracted from files stored on the target system. This information is passed back to the botnet herder via the C&C servers. The motivation is usually financial gain, but some types of information may be used for political purposes (viz. the recent publication of President Assad's emails). Of military significance, the advanced persistent threat (APT) captures and extracts information over a long period, typically months or years. The motivation is usually economic, political, or military espionage. The final attack type is sabotage, in which information stored in or passing through the target system is altered, usually with the aim of deceiving the user.

Threats and responses

From a botnet herder's viewpoint, the threat landscape is as complex as in counter-insurgency. The herder has to contend with several opponents. The first class of opponent is the target's developer and/or system administrator. The developer's choice of hardware, operating system, application software, and communications determine the herder's freedom of action. The system administrator can defeat a potential attack by putting strong defences in place and by responding promptly and appropriately to an attack attempt. Defences include firewalls (FW), anti-virus (AV) software, an intrusion detection system (IDS), and logging facilities. An equally important aspect of defence is the security policies and procedures that the system administrator has put in place. If the password policy is lax, then the target can be easy to penetrate. Similarly, neglect of updating and maintenance can make the herder's job easy by providing hardware or software vulnerabilities that

can be exploited. Defeating the system administrator is the botnet herder's first priority, and is a matter of intellectual enjoyment.

The second class of opponent is the developer of operating system, FW, AV and IDS software. Botnet herders are engaged in an arms race with these developers. As soon as a herder discovers a vulnerability in one of their products, the developers must implement a patch and distribute it to their clients. The most sophisticated botnet construction kits include facilities for encrypting several versions of bot software so that they pass "under the radar" of the leading AV products. The botnet herders then deploy each version of their bot software at a rate just faster than developers can field new patches and AV definitions. This is a classic example of using tempo to defeat the opponent's Observe-Orient-Decide-Act (OODA) loop (Boyd, 1996).

It may be surprising to learn that an important third class of opponent is the rival botnet herder. A botnet is a major asset that can generate tens of thousands of United States (US) dollars income per month. Building up a botnet can be a laborious business, and it may be less effort to take over all or part of another herder's botnet. This explains why one of the first thing bot software does after penetrating a target system is to search around for the presence of other bot software. If other bot software is found, it is removed or disabled. While there is a sense of community amongst botnet herders, there is also a pecking order with "script kiddies" at the bottom. They may exchange techniques, software, and knowledge about certain targets through dark fora and chat channels, but they keep their "crown jewels" for themselves. By contrast, law enforcement authorities come a poor last as a threat.

Botnet topologies and techniques

Botnets have evolved through three generations: open-source, kit-based, specialized (Czosseck, et al, 2011). Initially, botnets were written for fun or out of competition between hackers. Some of these botnets are still in use. They are easy to set up, typically just requiring configuration and compilation. Subsequent generations were mainly developed as an effective way of making money. In the second generation, botnets became available as construction kits, enabling herders to create their own botnet by point-and-click. A market in botnet software emerged, with a split in specialisation between botnet developers and botnet users. In this paper, we elide the two specialisations into one: the botnet herder. The third generation of botnets are those developed for a specific target or functionality, and may be used for espionage, sabotage, or as APTs. Specialized botnets are professionally developed, sometimes combining knowledge of the target domain with malware knowledge.

At the same time, botnet topologies changed from centralised control, often using Internet Relay Chat (IRC) as the communication channel, to decentralized, peer-to-peer (P2P) networks (Li, et al, 2009) (Zeldanloo & Manaf, 2009) (Wang, et al, 2010). This change parallels the change in military C2 from hierarchical to edge organizations. Networked botnets exhibited a variety of network models. There are botnet organized as random graphs, as small worlds, and as scale-free networks (Dagon, et al, 2008). The random-graph model avoids creating predictable flows, but it requires a central record of bots. This makes takedown of the botnet easy if a single bot is detected and analysed. Small-world botnets are constructed by new bots receiving a fixed number of prior zombies, typically around ten. This limits the damage if a single bot is detected and analysed. When botnets are constructed by preferential attachment, then a scale-free network model arises. Scale-free networks

are robust in the face of random detection and takedown, but vulnerable if the high-degree or high-bandwidth nodes (e.g., the C&C servers) are targeted.

Most botnets use the IRC protocol for communication because IRC server software is easy to obtain and to set up. However, botnets are relatively easy to detect from IRC traffic. Recently, botnets have been observed that use other technologies for their communications channel, including HTTP and very recently social networks such as Twitter (Kartaltepe, et al, 2010). The advantages of HTTP and Twitter for the botnet herder is that it is easier to hide botnet traffic "in plain sight". Even if botnet communication is detected, the identity of the C&C servers and bots and the message contents can be obscured by means of stealth and obfuscation techniques (Damballa, 2009a/b). Examples of these techniques include encryption, onion routing, lookup resilience, calling home, intermittent C&C, redundant C&C servers, IP and domain fluxing. Bot detection by AV software can be hindered by creating multiple varieties of the same bot and/or using multiple creator kits.

Comparison

Table 1 summarizes the comparison between military C2 and botnet C&C.

Table 1. Comparing military C2 and botnet C&C.

	Military C2	**Botnet C&C**
Closed loop	Yes (OODA)	Yes
Tempo	Yes	Yes
Domains	Physical, information, cognitive, socio-organizational	Information (cyberspace), cognitive
Supervisory control	Commander	Botnet herder
Direct control	Command team & C2 system	C&C servers
Process under control	Subordinate units	Zombie-hosted bots
Targets	Opposing forces	Target systems
Environment	Physical terrain, natural and man-made objects, weather, etc.	Open Internet
Infostructure	Dedicated; air-gapped from Internet	"Dirty" Internet
Organizational structure	Hierarchy (industrial age C2), transforming towards Edge organization (information age C2)	Centralized (hierarchy / scale-free); peer-to-peer (small worlds); unstructured (random graphs)
Threats	Opposing forces	System administrators, FW/AV/IDS developers, other botnet herders, law enforcement

Responses: information security techniques	Air-gapping, classification scheme, security principle (eg "need-to-know"), policies, procedures, encryption	Stealth & obfuscation techniques: encryption, onion routing, lookup resilience, calling home, intermittent C&C, redundant C&C, IP and domain fluxing, creating multiple varieties of bot, using multiple creator kits
Legal constraints	National and international law, national boundaries, jurisdictions	None
Physical constraints	Natural, national & man-made barriers; geography; metric distance; maximum speed of vehicle	None, except for diurnal rhythm of some zombie and target system users

The first question we must answer is whether military C2 and botnet C&C are the same kind of process. We observe that they are both closed loop, and both emphasize tempo. Both have an overall commander whose intent is communicated to and fulfilled by subordinates. The difference is that the subordinates in C2 are military units, while in botnet C&C they are fully automated bots. In this respect, botnet C&C is similar to industrial process control. However, botnet C&C distinguishes itself from process control in that it contends with intelligent adversaries with their own OODA loop, as in military C2. Hence, we conclude that the C2 and C&C processes are the same in character, although the military and botnet domains differ.

The botnet herder has an easier job than his military equivalent. Botnets operate only in the information and cognitive domains, with the herder being free of legal and physical constraints. In some circumstances, he must take into account the diurnal habits of the users of zombies and targets. If the user switches off his/her computer when he/she sleeps or frequently disconnects the computer from the Internet (e.g., when travelling), then this limits the botnet's actions. Given that the herder probably has many other bots and targets available, this is only a minor inconvenience. It can become a major concern when the botnet herder is engaged in a targeted attack, and the target user's computing is only intermittently accessible. As hackers' writings show (e.g., Mitnick & Simon, 2005), patience may then be an essential virtue.

The botnet operates in a "dirty" environment. The herder is forced to operate over the open Internet to access his C&C servers and bots and to approach his targets. Like other Internet users, he must secure his assets from malware. While he can exist guerrilla-style, living "off the land" in terms of taking over zombies as needed, he must "hide in plain sight". One special risk that he runs is having his botnet stolen from him by a rival botnet herder. For this reason, botnet herders have developed a range of stealth and obfuscation techniques that are rarely found in present-day military C2 systems. If the massive volume of innocent Internet traffic within which the botnet communications are embedded was to fall away, then the botnet would be dangerously exposed.

By contrast, military C2 systems primarily depend on air-gapping and encryption for protection against unauthorised access. Hiding "in plain sight" does not apply. However, air-gapping is leaky. As

for other computer systems, the operating system and applications software of C2 systems must be updated regularly. Cyber weapons and other malware can be introduced together with the updates. The greatest danger comes from the authorised user, who may be careless and – intentionally or unintentionally – violate the security policies and procedures. Wikileaks' publication of US State Department cables shows what can happen. Some authors (Schneier, 2000) (Mitnick, 2002) regard the insider threat as substantially greater than the threat of intrusion. Research in other areas (e.g., civil aviation and the petrochemical industry) shows that people deliberately violate procedures when they think that they know a better way of completing a task (Helmreich, 2000) (Hudson, et al, 1998).

A major disadvantage of air-gapping is that military C2 system users are denied access to the wealth of information and services available on the open Internet. That there is a pent-up need for Internet access is demonstrated in every military control centre where there is CNN feed. Social network sites are becoming an essential source of intelligence information. Internet access is vital in Information Operations, both in acquiring information and in influencing groups and key individuals. In humanitarian crisis response and management, where air-gapping is less prevalent, information obtained from social media is increasingly being merged with officially-sourced information (Bajpal & Jaiswal, 2011).

Suppose that military C2 system designers were to adopt the techniques that botnet herders already use. What advantages would this bring? In answering this question we will consider only the technical possibilities, leaving aside other issues, such as legal constraints, for future research. Moreover, we will assume that the full range of herders' stealth and obfuscation techniques is adopted.

The first advantage is that military C2 could be more able to "live off the land" with respect to information and communication technology (ICT) infrastructure. In terms of ICT resources, school and university computer laboratories would make good makeshift C2 centres, as would the premises of commercial companies that perform mainly administrative work (e.g., insurance companies).

A second advantage, closely linked to the first, is that the logistics footprint for setting up and maintaining military C2 systems could be reduced. Onsite ICT resources could be supplemented by highly portable, commercial off-the-shelf products, such as mobile phones, smartphones, and tablet computers. Instead of transporting C2 system hardware and software to the onsite location and then manning the system, the manpower would be issued with C2 system components in the form of Bring Your Own Device (BYOD), taking these components with them when deployed. After arrival onsite, they would simply join the growing C2 network.

A third advantage is that military C2 could be seamlessly informed from non-military sources. This would be more extensive than a CNN feed and intelligence information gleaned from social network sites. For example, it would enable C2 users to perform searches for information on the complete Internet using their favourite search engine. Users could also consult public databases (e.g., Wikipedia, YouTube, Flickr, etc.) and even access scientific publications (e.g., ScienceDirect, JSTOR, Web of Science, IEEE, etc). C2 systems themselves could be partly wholly implemented as mash-ups of services already existing on the Internet, like Google Earth and Google Maps.

A fourth advantage is that, in a complex endeavour, the interoperability with other military and non-military partners would be greatly simplified. Government departments, international organizations, commercial suppliers, and NGOs are invariable Internet-connected and use *de facto* standard applications. Communication and mutual understanding would be improved by being able to collaborate using chat / instant messaging, microblogging, (video) teleconferencing, and shared social network sites.

Conclusions and Recommendations

The C2 community has the ambition of developing information-age C2 systems that exploit the power of peer-to-peer networked collaboration at the edge of the organization. This paper has shown that much of this ambition has already been realized in the botnet herder community. The controllers of networks of compromised computers – botnets – have evolved a range of techniques for "hiding in plain sight" over the open Internet. We have made a *prima facie* case that these techniques could be adopted in military C2 systems, replacing the current practice of air-gapping. The associated operational advantages would include a reduction in C2 hardware and software procurement by "living off the land" of existing ICT infrastructure, bringing with it a reduction in the logistics footprint for C2 systems. Additional advantages include the ability to access public-domain information and services seamlessly, and improved interoperability with other partners in a complex endeavour.

Of course, what this paper offers is merely a concept at this stage. In terms of Technology Readiness Levels (TRLs) (Mankins, 1995), this concept is only at TRL 1 ("basic principles observed") or 2 ("technology concept and/or application formulated"), although the botnet herders' operational use of these techniques could be regarded as TRL 6 ("demonstration in a relevant environment"). The next step would be to perform a military C2 proof-of-concept analytically and experimentally to reach TRL 3. Given the nature of the concept, an obvious approach would be for interested parties to collaborate in an open- or crowd-sourcing manner.

References

Alberts, et al, 1999	Alberts, D.S., Garstka, J.J., & Stein, F.P. 1999. Network Centric Warfare. US Department of Defense Command & Control Research Program, Washington D.C.
Alberts & Hayes, 2003	Alberts, D.S. & Hayes, R. 2003. Power to the Edge: Command Control in the Information Age. US Department of Defense Command & Control Research Program, Washington D.C.
Bächer, et al, 2008	Bächer, P., Holz, T., Köter, M. & Wicherski. 2008. Know Your Enemy: Tracking botnets. The Honeynet Project, 8 October 2008. http://www.honeynet.org/papers/bots/, accessed 29 December 2011.
Bajpal & Jaiswal, 2011	Bajpal, K. & Jaiswal, A. 2011. A Framework for Analyzing Collective Action Events on Twitter. Proceedings, 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM), May 2011, Lisbon, Portugal.

Boritz, 2004	Boritz, J.E. 2004. IS Practitioners' Views on Core Concepts of Information Integrity. Technical Report, 30 September 2003, revised 12 March 2004, University of Waterloo Centre for Information Systems Assurance, Canada.
Boyd, 1996	Boyd, J.R. 1996. The Essence of Winning and Losing. Unpublished lecture notes, Maxwell Air Force Base, AL.
Czosseck & Podins, 2011	Czosseck, C. & Podins, K. 2011. An Usage-Centric Botnet Taxonomy. NATO Cooperative Cyber Defence Center of Excellence, Tallinn, Estonia.
Czosseck, et al, 2011	Czosseck, C., Klein, G. & Leder, F. 2011. On the Arms Race around Botnets: Setting up and taking down botnets. Proceedings, 3rd international conference on Cyber Conflict (CyCon 2011), NATO Cooperative Cyber Defence Center of Excellence, Tallinn, Estonia, 107-120.
Dagon, et al, 2007	Dagon, D., Gu, G., Lee, C.P. & Lee, W. 2007. A Taxonomy of Botnet Structures. Proceedings, ACSAC 2007.
Damballa, 2009a	Damballa. 2009a. Botnet Communication Topologies. White paper.
Damballa, 2009b	Damballa. 2009b. The Botnet vs. Malware Relationship. White paper.
Dittrich & Dietrich, 2008a	Dittrich, D. & Dietrich, S. 2008a. New Directions in Peer-to-Peer Malware. Proceedings, Sarnoff Symposium, IEEE.
Dittrich & Dietrich, 2008b	Dittrich, D. & Dietrich, S. 2008b. P2P as Botnet Command and Control: A deeper insight. Proceedings, Malicious and Unwanted Software, IEEE.
Galbraith, 1974	Galbraith, J.R. 1974. Organization Design: An information processing view. Interfaces, 4, 3, 28-36 (May 1974).
Galbraith, 1977	Galbraith, J.R. 1977. Organizational Design. Addison-Wesley.
Galbraith, 2008	Galbraith, J.R. 2008. Organization Design. In Cummings, T.G. (ed). 2008. Handbook of Organization Development. Sage Publications Inc., chapter 18, 325-352.
Grant, et al, 2007	Grant, T.J., van Fenema, P., van Veen, M.J.P. & Neerincx, M. 2007. On Regarding 21st Century C2 Systems and their Users as Fallible ePartners. Proceedings, 12th International Command & Control Research & Technology Symposium (ICCRTS 2007), Newport, RI, USA, 19-21 June 2007.
Grant, et al, 2011	Grant, T.J., Buizer, B.C. & Bertelink, R.J. 2011. *Vulnerability of C2 Networks to Attack: Measuring the topology of eleven Dutch Army C2 systems*. In Alberts, D.S. (ed.), Proceedings, 16th International Command & Control Research & Technology Symposium (ICCRTS11), June 21-24, 2011, Quebec, Canada, US DoD CCRP, paper I-087.

Helmreich, 2000	Helmreich, R.L. 2000. On Error Management: Lessons from aviation. British Medical Journal, 320, 781-785.
Hudson, et al, 1998	Hudson, P.T.W., Verschuur, W.L.G., Parker, D. Lawton, R. & van der Graaf. 1998. Bending the Rules: Managing violation in the workplace. Invited Keynote Address, Society of Petroleum Engineers, 1998 international conference on Health, Safety and Environment in Oil and Gas Exploration and Production (SPE 1998), Caracas, Venezuela.
Kamluk, 2008	Kamluk, V. 2008. The Botnet Business. White paper, SecureList, May 2008.
Kartaltepe, et al, 2010	Kartaltepe, E.J., Morales, J.A., Xu, S. & Sandhu, R. 2010. Social Network-Based Botnet Command-and-Control: Emerging threats and countermeasures. Proceedings, ACNS 2010, Springer-Verlag, Berlin, LNCS 6123, 511-528.
Leder, et al, 2009	Leder, F., Werner, T. & Martini, P. 2009. Proactive Botnet Countermeasures: An offensive approach. In Czosseck & Geers (eds). The Virtual Battlefield: Perspectives on Cyber Warfare, IOS Press.
Li, et al, 2009	Li, C., Jiang, W. & Zou, X. 2009. Botnet: Survey and Case Study. Proceedings, 4th international conference on Innovative Computing, Information and Control, IEEE, 1184-1187.
Mankins, 1995	Mankins, J.C. 1995. Technology Readiness Levels: A white paper. NASA Advanced Concepts Office, Office of Space Access and Technology, Washington DC, USA.
Mintzberg, 1980	Mintzberg, H. 1980. Structure in 5's: A synthesis of the research on organization design. Management Science, 26, 3, 322-341 (March 1980).
Mitnick & Simon, 2002	Mitnick, K.D. & Simon, W.L. 2002. The Art of Deception: Controlling the human element of security. Wiley Publishing Inc, Indianapolis, IN, USA.
Mitnick & Simon, 2005	Mitnick, K.D. & Simon, W.L. 2005. The Art of Intrusion: The real stories behind the explits of hackers, intruders & deceivers. Wiley Publishing Inc, Indianapolis, IN, USA.
Monsuur, et al, 2011	Monsuur, H., Grant, T.J., & Janssen, R.H.P. 2011. *Network Topology of Military Command & Control Systems: Where axioms and action meet*. In Bauer, J.P. (ed.), Computer Science, Technology, and Applications, vol 3, pp. 1-27. Nova Science Publishers, Inc., Hauppauge, NY, USA.
Nissen, 2005	Nissen, M.E. 2005. Hypothesis testing of edge organizations: Specifying computational C2 models for experimentation. Paper presented at the 10th International Command & Control Research Symposium, in McLean, VA.
Orr & Nissen, 2006	Orr, R.J. & Nissen, M.E. 2006. Hypothesis testing of edge organizations: Simulating performance under industrial era and 21st century conditions.

	Paper presented at the 11th International Command and Control Research and Technology Symposium in Cambridge, UK.
Puri, 2003	Puri, R. 2003. Bots & Botnets: An overview. White paper, SANS Institute.
Schneier, 2000	Schneier, B. 2000. Secrets and Lies. John Wiley & Sons.
Wang, et al, 2010	Wang, P., Aslam, B. & Zou C.C. 2010. Peer-to-Peer Botnets: The next generation of botnet attacks. Handbook of Information and Communication Security, Springer, Part C, 335-350.
Zeldanloo & Manaf, 2009	Zeldanloo, H.R. & Manaf, A.A. 2009. Botnet Command and Control Mechanisms. 2nd International Conference on Computer and Electrical Engineering (ICCEE'09), 564-568.