CHAPTER

10

Computer Network Attack

INFORMATION IN THIS CHAPTER

- Waging War in the Cyber Era
- The Attack Process

Computer Network Attack (CNA) is a military term defined as, "Actions taken through the use of computer networks to disrupt, deny, degrade, or destroy information resident in computers and computer networks, or the computers and networks themselves" [1]. Although this term meshes well with the common viewpoint of basements full of hackers bringing cyber war to the enemy, or individual attackers conducting similar activities, there is a large difference in how such activities are conducted by nation-states and non-nation-states.

It is entirely true that, in a purely cyber war sense, small groups or individual attackers can potentially wield similar weapons to a similar level of effectiveness as a nation-state, but the similarity will often end there. An individual hacker with access to the command and control system of a large botnet can certainly wreak havoc, but the capability to take the attack into conventional warfare, or to use the cyber attack as an accompaniment or complement to other attacks is often reserved for those with much greater resources.

Another common confusion when discussing CNA is differentiating it from the common attacks carried out by blackhat hackers and other similar groups that are not being actively sponsored by a nation-state, or even in the attacks that we carry out against ourselves in the penetration testing process. The difference, primarily, is a matter of scale in capabilities and completeness of the attack process.

Attacks conducted in the name of penetration testing and by random hackers do not usually "go for the throat" as we might in a conventional attack. Many such attackers work to compromise the target environment in order to own it, but do not take the destructive steps beyond that which might be required in actual warfare. In full-blown cyber warfare, where we have a presumably greater intent to significantly impact our target, such steps might lead to the wholesale destruction or disabling of critical infrastructure through a purely cyber

http://dx.doi.org/10.1016/B978-0-12-416672-1.00010-6

attack, or might be used to disable systems that provide protection against a conventional attack, such as missile tracking systems, in order to facilitate such an attack.

WAGING WAR IN THE CYBER ERA

Cyber warfare capabilities are not only relatively new, when discussing them on their own merits, but they change the way conventional warfare is carried out as well. When we look at any of the traditional methods of warfare, cyber capabilities add new dimensions to them. In cyber warfare, we must consider the physical, electronic, and logical elements of warfare as major factors, as well as the reasons for our actions and the factor of time.

Physically

Cyber warfare can have great impact on the way physical war is waged. Given that even strictly physical warfare, in the sense of boots on the ground, depends a great deal on technologies, these things are vulnerable to cyber attack. Support for physical operations depends on supplies being delivered properly, soldiers being moved from one place to another on a tight schedule, communications functioning, and any number of other factors, e.g., the Army's six warfighting functions. If one or more of these activities does not take place, or, worse yet, is intentionally altered in order to engineer a weakness, solely physical warfare can quickly degenerate into chaos.

On the other side of the coin, cyber warfare activities are very vulnerable to physical effects. If communications lines are severed, power is unavailable, environmental conditions cannot be maintained, or any of a number of other conditions cannot be met, our relatively fragile computer systems and infrastructure become so much dead weight.

In either case, physical warfare can affect or be affected by cyber warfare attacks. When the physical component is ignored in cyber warfare, we potentially lose a large portion of the entire picture. Cyber warfare is indeed a distinct dimension of warfare, but isolating it from the other dimensions renders its capabilities incomplete, at best.

Electronically

Although often considered a subset of conventional or physical warfare, electronic warfare can have a profound effect on cyber warfare and vice versa. Electronic warfare is largely concerned with attacks that take place in the electromagnetic spectrum, an area which the systems that are used to carry out cyber warfare make great use of, and from which they are very sensitive to interference. Using the tactics of electronic warfare, we can potentially render the systems and infrastructure that make up the cyber warfare capabilities of our opponents useless without landing a single physical blow.

Likewise, the systems that allow electronic warfare to be carried out are generally of a highly technological nature and are potentially susceptible to attack on a cyber level. One can envision an exchange where a nation-state would attempt to remove the cyber capability

from an opponent via electronic warfare attack, only to find that its electronic warfare capability had been nullified by a cyber attack.

Logically

Of course, as we discussed in the introduction to this chapter, we also have strictly cyber oriented attacks to consider. Such attacks can be used for reconnaissance and surveillance, as we discussed in Chapter 9, but they can also be used to conduct outright attacks against other systems and infrastructure. Such attacks are the meat of CNA and we will spend a considerable amount of time discussing them later in this chapter.

Purely logical attacks in isolation are very much lacking in their potential to be effective in an overall war effort. Although it is very easy for nearly any party to obtain and utilize such weapons to great effect, not being able to follow up with other attacks is extremely limiting. If we consider conflicts of a conventional nature as an example, using cyber warfare tactics in isolation might be the equivalent of conducting conventional warfare without the use of air support; definitely possible, but very limiting.

Reactively Versus Proactively

In considering cyber warfare attacks, we can act reactively, in the sense of defending against an attack or responding to the actions of our opponents. We can also act proactively, in the sense of anticipating activities stemming from threats or courses of action on the part of our opponents that would seem to indicate progress toward an undesirable state. Given cyber capabilities, we have the possibility of using tactics that are not immediately physical or overtly harmful, and do not require physical movement of troops or resources to carry out such activities.

When responding reactively, we will likely continue in the paradigm of traditional warfare. Although we do not necessarily need to move resources into the area, we still need to conduct many of the staging operations that are required to ramp up for such a conflict. In all likelihood, this will include conducting many of the reconnaissance activities that we discussed in Chapter 9 when discussing Computer Network Exploitation (CNE), and may be able to benefit from any ongoing surveillance that was already in place against our target. Once such activities are completed to the extent that we have sufficient information to conduct attacks, we can then move on to CNA.

If we are to conduct cyber warfare proactively, we have a very large range of options that are technically open for use, up to and including an all-out attack. At present many countries limit the use of such options via internal legislation and international treaty. Of great potential usefulness, however, are attacks that are put in place in advance, but not triggered until conditions are the most appropriate and advantageous for us to do so. Such tactics can be staged years in advance and may even be insinuated into the systems of our opponent at a hardware level. We discussed such activities in greater depth in the Supply Chain Concerns section of Chapter 7. In such situations, carefully planned proactive activity can be used to render the opponent entirely impotent at the exact time in which they are most dependent on their tools and weapons to function properly.

Time as a Factor

When conducting cyber warfare, we have the capacity to unleash an attack at speeds which are far above the reaction times of mere humans, presuming that humans are not a requirement in the decision-making loop. We may see actions take place on such a time scale in the operation of Intrusion Detection Systems (IDS) that are defending our systems, or in the autonomous or semiautonomous strikes from attack tools, which we will discuss at greater length in Chapter 12. Although such attacks are entirely possible to carry out in very short periods of time, they do not accurately represent the entirety of cyber warfare and Computer Network Operations (CNO), any more than an individual soldier firing a weapon represents an entire war.

Pure cyber warfare on a grand scale, which we have not yet witnessed, will, in the opinion of the authors, likely be a relatively slow operation. It will be prefaced by similar reconnaissance and surveillance activities that we see in conventional warfare, and, in fact, will probably be accompanied by conventional warfare activities. In small-scale skirmishes, such as the force of a major botnet being directed against government systems, we may not see the full engine of warfare brought to bear against the attackers, and in this particular case, we may see a swift and cyber only attack. In all likelihood, the speed at which entire conflicts are fought will closely model that of conventional war.

THE ATTACK PROCESS

The attack process is usually focused on a particular system or set of systems. In this process, as shown in Figure 10.1, we will likely conduct additional and more detailed reconnaissance and scanning oriented toward gaining yet more specific information from the system. At this level, we can potentially conduct reconnaissance in greater depth, as our need for secrecy and stealth may not be as great as it was while we were conducting CNE. We will then attempt to access the system, either through the use of an outright attack or using credentials that we have managed to gather from somewhere in the environment, through social

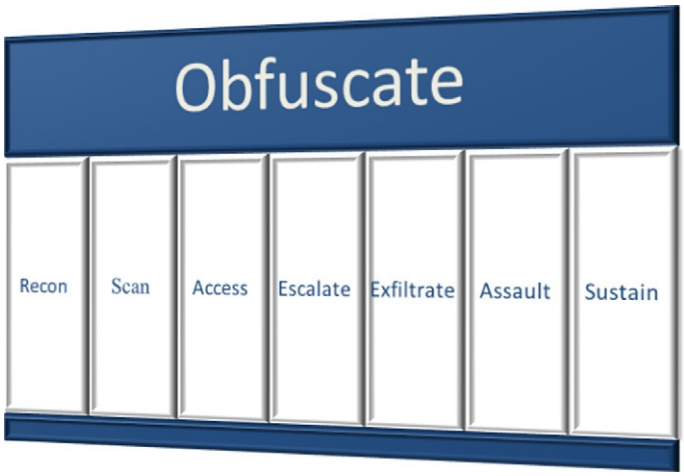

FIGURE 10.1 Attack process.

engineering, or other means. Once we have an account on the system, we may need to escalate the level of access that we have in order to accomplish our goals. The target for such privilege escalation is often root or administrator level access, giving us relative freedom on the system. Given the needed level of access to the system, we can then exfiltrate any information that we wish to, cause damage to the environment in any way that benefits us, then install any measures that we need to in order to ensure future access.

Throughout the entire attack process, we will also seek to cover or obfuscate our activities. We may want to appear to be attacking from a different location than where we are physically located or take other steps to ensure that our attacks are not traced back to us. We will also likely wish to remove any traces of our activities on the system when we intend to leave it.

Recon

We spent a good deal of time discussing reconnaissance and surveillance in Chapter 9 in the context of CNE. In that case, the reconnaissance that we would conduct would be done in a general sense, in order to map out and discover information on our target environment. As reconnaissance done in support of CNA and of the attack process, we may already have such general information already from the CNE and will be hunting for information on a much more specific level, given our potentially greater level of access and reduced need for stealth.

Another tool that may become useful during this more specific stage of reconnaissance is social engineering. Using some of the social engineering tactics that we discussed in Chapter 8, we may very well be able to gain specific information that will allow us to access the systems in question without needing to resort to the full spectrum of attacks that we might need otherwise. Through social engineering we may be able to discover shared passwords used in other services or applications, may be able to find account names through searching the physical surroundings of those that work in the environment or through dumpster diving, or any number of similar tactics.

Given the task of long-term reconnaissance at a more specific level, we may also want to plant the tools that would allow such monitoring on a particular system. Even on this scale, software such as a keystroke logger can produce enormous amounts of information, only a very small portion of which will generally have any great value; however, it may still be worth the effort. In environments where good password hygiene is not strictly enforced with technical controls, we can often find passwords that are manually synchronized between multiple systems, a great boon when attempting to gain access. We may also be able to sniff credentials from network traffic if less secure protocols such as telnet, File Transfer Protocol (FTP), or Post Office Protocol (POP) are allowed in the environment.

TIP

We should be prepared, at any step in the attack process, for our attacks to fail utterly and/or to be discovered. Particularly when our target is a highly secured environment, and we are facing stronger measures, such as multifactor authentication, this may very well be the case. It is always wise to have contingency plans that will allow us to still achieve our goals when we encounter such obstacles.

Scan

During the scanning portion of CNA, instead of the general port scans, fingerprinting, service versioning, and so on that we performed in our general reconnaissance, we will likely more closely examine the system for potential vulnerabilities during reconnaissance in CNA. In general, we will be scanning for further detailed information from applications and potentially more specific information from the operating system itself.

When attempting to collect more information from applications, beyond cursory checks for versions, we will often focus on finding an exposed application that might be particularly talkative, such as a web interface to a database, and drilling down from there. This is often a manual process and can be time consuming, but can be very useful. We can often discover very specific information in this manner, such as database versions from error messages, potential usernames from conducting SQL injection attacks through the web interface, and any number of other bits and pieces of information.

NOTE

Not only can applications provide us an opportunity to surveil a remote system, but they can also potentially provide us an open doorway into the operating system itself. Improperly secured web applications are one of the main vectors that allow such attacks to take place.

We may also want to collect additional information regarding the operating system such as specific patching information, uptime, or any of a number of other items that could potentially allow us to gain information through inference. Such additional small details may aid us in our attacks when we get to the attack and escalation steps of our process. As we discussed in the more general information collection sections of Chapter 9, documenting this information carefully can be very helpful through the entire process.

Access

Gaining access to a system can take place using a variety of tools and methods. If we have been successful in any of our previous attempts at social engineering, dumpster diving, stealing, or cloning access cards, such as Common Access Cards (CACs) mandated by Homeland Security Presidential Directive 12: Policy for a Common Identification Standard for Federal Employees and Contractors or have managed to find accounts with synchronized passwords on other systems that we have been able to access, we may very well have legitimate credentials with which we can simply log in. Slightly more complicated than this, although more likely, is that we will be able to find usernames that exist on the system and either crack or guess passwords, using some of the tools that we discussed in Chapter 6, in order to access them.

Another potential path that may gain us easy access would be to use client-side attacks against individual systems that belong to the users of our target system. Such attacks utilize vulnerabilities in software running on the client, such as a web browser, as an attack vector. We stand a much greater chance of being able to access individual workstations in order to gain access to credentials than we do when attempting to access a server that is carefully

maintained and patched. Client-side attacks can be web-based, use email as a delivery method, ride in on a USB drive, or any of a number of other methods. One example of this that is commonly known today is the 2008 cyber attack on U.S. military computers in history named "Operation Buckshot Yankee." The case involves USB flash drives infected by a foreign intelligence agency and prompted the military to ban the use of them [2]. Particularly in nontechnical working environments, such attacks enjoy a high degree of success, although we may not find as much success in highly secured environments.

TIP

Client-side attacks are often some of the most effective attacks that we can carry out. Such attacks, when combined with a certain element of social engineering, as we discussed in Chapter 8, are very difficult to defend against. When we use human carelessness or ignorance as an attack vector, we will often enjoy success.

We can also attempt to use common operating system or application exploits in order to access a system. We have likely, at some point in the process, already used one or more of a variety of vulnerability scanning tools, either during the more general reconnaissance process, or during the more specific examination during the attack process.

NOTE

In the case of a cyber attack, we will likely not use exploits that are available to the general public. Such vulnerabilities are likely to already be patched or mitigated in some fashion, and easily rebuffed. Instead, we will use zero-day exploits which stand a much greater chance of success, due to not being commonly known.

Many common vulnerability analysis tools, such as Nessus, which we discussed in Chapter 6, can be used to locate vulnerabilities that we might use to access a system. Although it is unlikely that we will gain access in such a fashion on a fully patched system running a recent operating system, there are plenty of systems that are likely not in such a well-maintained state to which we may easily be able to gain access. It is also important to test our attacks in an environment as close as possible to the actual target as we can create. This will allow us to not only test our exploits, but to also help develop contingency plans to potentially compensate for issues that we might encounter when attacking.

Escalate

Once we have gained some sort of access to a given system, we may need to gain additional or upgraded privileges than those that we presently have, commonly known as privilege escalation. When we are attempting to gain access to accounts that have a higher level of privilege than those that we presently have, this is known as vertical privilege escalation. When we are attempting to gain access to different accounts that what we have access to,

but are at the same level as the account that we already have access to, this is known as horizontal privilege escalation.

Privilege escalation of either variety can be accomplished through a variety of methods. We may be able to use a different set of exploits than we used previously, as we now have access to the system as a user. We may also be able to take advantage of misconfigurations or insecurely set configurations. It is entirely possible that, on some systems, the standard user account that we have managed to access may have the ability to act as an administrator directly or may be able to escalate privilege level as normal functionality of the operating system.

We may also be able to utilize the privileges of applications that are operating with heightened permissions. Applications such as those that run backups, various servers or daemons, or other processes that require privileges that are higher than the level of a general user are often vulnerable to attack. Various application flaws such as buffer overflows or race conditions can allow us to execute arbitrary code through these already running applications. We may also be able to access and modify interpreted scripts or shell scripts that are not secured properly, in order to pass operating system commands through them or gain direct access to an operating system shell.

Exfiltrate

Once we have gained the needed access to the environment, one of our primary concerns is to find any data that may be valuable to us, and exfiltrate it to a location that is accessible to us from another location, or to move it directly to our own systems. Exfiltration, in terms of Confidentiality, Integrity, and Availability (CIA), is an attack primarily against confidentiality, and potentially against availability.

We have a very wide variety of tools that we can use to exfiltrate data, from purpose-built tools and protocols that exist for the specific purpose of moving data around, to more general tools that can be bent to such a purpose, to out-of-band methods that might allow us to subvert security measures designed to specifically prevent such efforts.

In simple cases, we may be able to easily use common applications and protocols to move our files or data. File transfers can be accomplished with FTP, Secure Copy Protocol (SCP), Extensible Messaging and Presence Protocol (XMPP), or any of a number of other common protocols. In many environments we may find these particular transfer protocols blocked as outgoing traffic, but we will often find Hypertext Transfer Protocol (HTTP) traffic allowed, which will suit our purposes nicely. It is a rare and highly secure environment indeed where we will not be able to find some sort of outgoing protocol on which we can piggyback information.

In some cases, we need to create more specifically tuned tools in order to move our data out of the target environment in which we are operating. In such a case, netcat can be a very powerful tool for moving data around in a customized manner. The netcat tunnel setup that we discussed in multipurpose tools section of Chapter 6 would allow us to configure specific ports on each end of the connection and even relay the data through multiple systems in order to exfiltrate it from a hostile environment.

Assault

Assault is a step typically not included in the penetration testing process, which, in general, closely mirrors our attack process. In the case of actual cyber warfare, it is likely that once we have managed to gain access to a machine, escalate to the privilege level that we need, and exfiltrate any interesting data, we may want to use the system to sow chaos in the environment. In military terms, we have the five Ds to describe the effect of such activities: deception, disruption, denial, degradation, and destruction [3], as shown in Figure 10.2. In a CIA sense, the attacks in this section will mainly be against availability and integrity.

Deception is a somewhat more subtle tactic that carrying out a forthright attack, such as simply taking down a given system. If we can take over a system responsible for the control of communications, such as an email or Voice over IP (VoIP) server, we have the potential to falsify communications, alter those that are in transit, or simply make such traffic disappear while en route. The potential for internal manipulation in this way is relatively endless, as we, as a society, are highly dependent on such tools for communication and generally very trusting of them. Rendering them untrustworthy can be very psychologically destabilizing for the users of the target systems. In very high security environments, we may be hampered in such efforts by encryption or other similar verification systems, but, in such a case, we may merely need to insinuate ourselves into other sources of information. In the modern world, the computer is king and is not often questioned.

Causing disruption in systems that are on or connected to a computer is often a relatively easy proposition. On the systems themselves, processes can be interrupted, resources can be over-utilized, files can be moved or manipulated, or any number of other similar tasks. Particularly when timed to coincide with predictably scheduled tasks, such a system patching, quarter or year end financial closing, or large-scale military operations, system disruptions can cause panic and disruption among users and system administrators that is far out of proportion with the actual events.

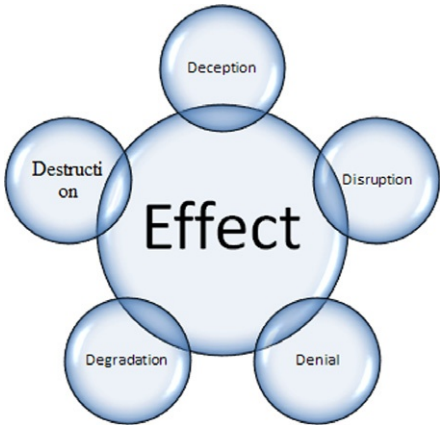

FIGURE 10.2 Five Ds.

WARNING

Carrying out an assault on a computer system may have implications far beyond the actual act itself. Particularly in the case of an attack launched by or attributed to a nation-state, such activities can lead to outright war, potentially including conventional warfare as a component.

Denial attacks of a certain variety are common enough in the world of information security. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are a frequent occurrence, largely due to the ease with which they can be executed. Such attacks are commonly launched against web servers, mail servers, FTP servers, and other public-facing components of a company or an organizational infrastructure. We can also launch such attacks against the technological components of physical access control systems, transportation systems, such as air traffic control, or any number of other critical components. In many complex systems, there are similar points at which failure can bring an entire process or facility to a swift halt.

Degradation in computing or industrial environments can be a virtual plague on those that use and maintain the environment. We can attack the performance of the system and networks, making portions of it function poorly in a sporadic manner, in order to cause those troubleshooting the issues to spend an inordinate amount of time on them. We can, in industrial systems, cause the output of the system to vary from its baseline, thus providing goods that are not produced to specification or are damaging to the infrastructure through which they move. We can also cause subtle degradation of data that is produced by a system, causing the results of medical tests to change, financial decisions to be altered, targeting systems to select incorrect locations, and no other end of harm that would be difficult to detect.

The term *destruction* can cover a wide variety of actions. We can destroy data, applications, or the operating system of the system in a software sense, which could potentially be very damaging in the case of valuable data or critical systems. We can attempt to physically damage the hardware that the system itself runs on, although this is a relatively limited tactic. Even in the case that we manage to destroy computing hardware, such equipment is often cheaply and easily replaced.

Sustain

Once we have gained sufficient access to a system, we may wish to reconfigure it to ensure our future ability to access it again. Although we may have used a specific exploit to gain access to the system and escalate our privileges when we were first able to do so, we may not be able to count on the same points of entry being available in the future. Against this eventuality, we will likely want to secure additional access by creating new accounts, opening services on additional ports, installing command and control software, placing backdoors in applications, and so on.

The most successful such efforts will likely be those that are the least obvious and the least prone to being accidentally discovered by a system administrator. Some of the more blatant methods, such as opening a new listening port on the system may very well be found in short order, particularly on an Internet-facing system. Additionally, we may want to be careful of leaving behind such measures in places where they might be found by another attacker. Many

of the prebuilt backdoors that are available will use a standard port by default, which could render our backdoor very easily located if we do not change it.

In addition to leaving backdoors in place, we may also want to consider, as an attacker, patching or fixing the vulnerabilities through which we were able to gain access. If such systems are left in their original state, we may find that another attacker has used the same methods and that we are now sharing control of the system. Worse yet, future attackers may not be as careful as we have been, thus revealing that the system has been compromised, triggering a further investigation and potentially severing our access in the process.

Obfuscate

Our likely first and last step on a system that we have compromised or intend to compromise is obfuscating, largely with the goal of anonymity or deniability. Obfuscate means "to confuse, bewilder, or stupefy" [4]. We use this term to cover not only the methods that we might use to cover up or erase evidence of our intrusion, but also to potentially point any potential investigators to another source entirely. Obfuscation is really a layer that runs under all of the activities that we will take in the attack process. Some such obfuscatory actions take place even before our first recon, some take place during our various attacks, and some take place as our very last step before permanently vacating the system in question.

The simplest and earliest obfuscation measures that we might take are those that will prevent our attacks from being traced back to our actual physical location. Such tools might be various proxies or intervening machines that we use as an intermediary connection before attacking, IP spoofing, or any of a number of other methods that we might use to disguise our point of origination. Although some such tools may not be perfect in nature, they do provide an additional layer of protection in case our activities in the target environment are noticed.

We will also likely take steps to ensure that we do not leave digital forensic evidence behind on the target system. In such cases, we might change timestamps so that they reflect the original time before we modified any files, clean up any tools that we have moved to the system, remove or alter log entries, and generally ensure that we have not accidentally left any traces behind. On the other side of this same process, we may very well want to intentionally leave such traces behind but alter them so that they point to another source. If we can falsely attribute an attack to another source, this may not only cover our tracks, but cause significant confusion and consternation as well.

SUMMARY

In this chapter we discussed CNA. We covered the different factors involved in cyber warfare, including the physical, logical, and electronic elements of warfare. We also covered reactive and proactive actions in warfare, and how these prompt a rather different set of actions in cyber warfare. Additionally, we must concern ourselves with the factor of time, and consider that although cyber attacks can be conducted very quickly, cyber warfare cannot be conducted at such speeds.

We also discussed the different phases of the attack process: reconnaissance, scanning, accessing systems, escalating privileges, exfiltrating data, assaulting the system, sustaining our access, and obfuscating any traces that might be left behind. We covered the specifics of each step in the process, and how some of the tools that we covered in Chapter 6 might be applied to each of them.

These processes and the tools that we have discussed outline some of the major strategies and tactics that are used to conduct CNA. These tools are not unique, nor are many of them difficult to access, and the process is simple, but to carry out warfare at the level of a nation-state requires a great deal more resources, effort, and knowledge.

References

[1] Joint Publication 1-02 Department of Defense Dictionary of Military and Associated Terms, http://ra.defense.gov/documents/rtm/jp1_02.pdf; 2011 [accessed 5.5.13].
[2] Lynn W. Defending a new domain: the Pentagon's cyberstrategy. Foreign affairs, http://www.foreignaffairs.com/articles/66552/william-j-lynn-iii/defending-a-new-domain; 2010 [accessed 20.05.13].
[3] U.S. Air Force. AFDD 2-5 information operations. s.l.: U.S. Air Force. Air Force Doctrine Document 2-5, http://www.globalsecurity.org/military/library/policy/usaf/afdd/2-5/afdd2-5.pdf; 1998 [accessed 18.03.13].
[4] Dictionary.com. Obfuscate. Dictionary.com, http://dictionary.reference.com/browse/obfuscate; 2013 [accessed 18.03.13].