2006-03

A Taxonomy of Deception in Cyberspace

Rowe, Neil C.

Monterey, California. Naval Postgraduate School

International Conference in Information Warfare and Security, Princess Anne, MD, March 2006
https://hdl.handle.net/10945/35976

A Taxonomy of Deception in Cyberspace

Neil C. Rowe
U.S. Naval Postgraduate School, Monterey, California, USA
ncrowe at nps.edu

Abstract: Deception is an important component of information operations, valuable for both offense and defense. We enumerate the space of possible deceptions using a new approach derived from semantics in linguistics, including some ?second-order? deceptions. We rate the appropriateness of each of the deceptions for offense and defense in cyberwar, and provide some detailed examples.

Keywords: Deception, taxonomy, cyberspace, case grammar, defense

This paper appeared in the International Conference in Information Warfare and Security, Princess Anne, MD, March 2006.

1.1 Introduction

Deception is a classic tool of military operations, and can often work as a potent ?force multiplier?. As battlespaces of the future increasingly involve cyberspace, we should explore what forms of deception apply there. Unfortunately, many analogies can be misleading for cyberspace, as identities and locations are more fluid and social interactions are quite different. Thus we need to carefully examine proposed analogies to develop a menu of tactics and strategies for deception planning for a military operation, either offensively or defensively. Computer systems and networks are being attacked all the time by "hackers" (Chirillo, 2002) and "social engineers" (Mitnick, 2002) so we already have evidence about what deceptions can work.

Moral objections can be raised to the deliberate use of deception. However, deception has many legitimate uses in human interactions (Nyberg, 1993). It has a long history in warfare (Latimer, 2001), and is a classic tactic and strategy for the more vulnerable party (Hutchinson & Warren, 2001). In cyberspace, technologically advanced countries like the United States are the most vulnerable so they may benefit more from deception.

2. Previous taxonomies

Several taxonomies of deception have been proposed. (Bell & Whaley, 1991) gives six categories in two groups of three: masking, repackaging, dazzling, mimicking, inventing, and decoying. All these have analogies in cyberspace:

- Masking (hiding in the background) occurs with "low and slow" attacks that are accomplished over a long period of time to avoid appearing suspicious.
- Repackaging (hiding as something else) occurs with Trojan horses that to do some concealed function in addition to their avowed purpose.
- Dazzling occurs with denial-of-service attacks that flood a target with a large volume of information at once.
- Mimicking occurs with ?phishing? that uses fake Web sites to steal personal data.
- Inventing occurs with honeypots, machines that invite attacks to collect data about attackers.
- Decoying occurs with scams of the ?Nigerian letter? type where victims are decoyed by the promise of future money.

Dunnigan and Nofi (2001) propose a taxonomy of military deception, most of which apply to cyberspace (Rowe & Rothstein, 2004):

- Concealment, essential to many attacks in cyberspace.
- Camouflage, done in Trojan horses.
- Disinformation, possible in cyberspace in files and messages for spies to find.
- Lies, useful since many statements about details of computer systems are hard to disprove.
- Displays, not too useful since a real system is usually easier to maintain than a fake one.
- Ruses, used frequently when attackers masquerade as legitimate users.
- Demonstrations, probably counterproductive in cyberspace since hackers like challenges.

- Feints, not useful for the same reason.
- Insight, possibly useful but requires considerable work to set up (Rowe, 2003).

3. A taxonomy from linguistic case theory

These taxonomies are insufficiently detailed enough to provide good guidance for constructing deception plans for cyberspace. So we have been investigating an approach based on linguistics ((DeRosis et al, 2003) provides an alternative formulation). Each action has associated concepts that help particularize it, and these are conveyed in language by modifiers, prepositional phrases, participial phrases, relative clauses, infinitives, and other constructs. These associated concepts are called ?semantic cases? (Fillmore, 1968) in analogy to the syntactic cases that occur in some languages for nouns.

Our claim is that every deception action can be categorized by an associated semantic case or set of cases. There is no canonical list of semantic cases in linguistics though systems for automated natural-language processing always use them. We prefer the detailed list from (Copeck et al, 1992), supplemented by two important relationships from artificial intelligence, the upward type-supertype and upward part-whole links, and two speech-act conditions from (Austin, 1975), to get 32 cases altogether:

- space
    - direction, of the action
    - location-at, where something occurred
    - location-from, where something started
    - location-to, where something finished
    - location-through, where some action passed through
    - orientation, in some space
- time
    - frequency, of occurrence of a repeated action
    - time-at, time at which something occurred
    - time-from, time at which something started
    - time-to, time at which something ended
    - time-through, time through which something occurred
- participant
    - agent, who initiates the action
    - beneficiary, who benefits
    - experiencer, who senses the action
    - instrument, what helps accomplish the action
    - object, what the action is done to
    - recipient, who receives the action
- causality
    - cause
    - contradiction, what this action opposes if anything
    - effect
    - purpose
- quality
    - accompaniment, an additional object associated with the action
    - content, what is contained by the action object
    - manner, the way in which the action is done
    - material, the atomic units out of which the action is composed
    - measure, the measurement associated with the action
    - order, with respect to other actions
        - value, the data transmitted by the action (the software sense of the term)
- essence
    - supertype, a generalization of the action type
    - whole, of which the action is a part
- speech-act theory
    - external precondition on the action
        - internal precondition, on the ability of the agent to perform the action

We can analyze the adequacy of the cases for cyber-warfare as follows, leaving quantitative rating for section 6. More examples from cyberwar for this taxonomy are provided in (Rowe and Rothstein, 2004).

3.1 Spatial cases

Actions have associated locations, and deception can apply to those references. However, a person cannot be said to inhabit cyberspace since they can simultaneously control more than one computer system, and packet routing through machines unknown to the attacker and defender is common on the Internet. It is thus not possible to deceive in ?location-at? or ?location-through?. Deception in ?location-from? or ?location-to? is possible since one can try to conceal one's location in launching or defending against an attack. Direction and orientation cases can arise with some actions that are supposedly one-way like file transfers.

3.2 Time cases

Computers can operate 24 hours a day without getting tired, so deception in time to enable a surprise attack or defense is not often possible in cyberspace, except when people play an important role in operations. However, many actions on computer are timestamped, and attackers and defenders can deceive in regard to those times. So an attacker could change the times of events recorded in a log file or the directory information about files to conceal records of their activities. Frequency is also an excellent case for deception, as in denial-of-service attacks that greatly increase the frequency of requests or transactions to tie up computer resources.

3.3 Participant cases

Actions have associated participants and the tools or objects by actions are accomplished. Identification of participants responsible for actions (?agents?) is a key problem in cyberspace, and is an easy target for deception. Deception in objects of the action is also easy: Honeypots deceive as to the hardware and software objects of an attack, and ?bait? data such as credit-card numbers can also be deceptive objects. The recipient of an action in cyberspace is usually the object. Deception is easy with the instrument case because details of how software accomplishes things are often hidden in cyberspace. Deceptions involving the beneficiary of an action occur with phishing and other email scams. Deception in the ?experiencer? case occurs with secret monitoring of adversary activities.

3.4 Causality clues

Deception in cause, purpose, and effect is important in many kinds of social-engineering attacks where false reasons like "I have a deadline? or "It didn't work" are given for requests for actions or information that aid the adversary. Deception in a contradiction action is not possible in cyberspace because commands do not generally relate actions.

3.5 Quality cases

The ?quality? semantic cases cover the manner in which actions are performed. Deception as to accompaniment and content is essential to planted disinformation and to Trojan horses that an adversary can manipulate. Deception in value (or subroutine "argument") can occur defensively as in a ploy of misunderstanding attacker commands. Deception in measure (the amount of data) is important in denial-of-service attacks and can also done defensively by swamping the attacker with data. Deception in material does not apply much because everything is represented as bits in cyberspace, though defenders can deceive this way by simulating commands rather than executing them. Deception in manner does not generally apply because the manner in which a command is issued or executed should not affect the outcome. Similarly, the order of commands and events can rarely be varied and even then cannot easily deceive anyone.

3.6 Essence cases

Deception can occur in the ontological features of an action, its type and the context to which is belongs. Phishing email is

an example of deception in supertype, where what appears to be a legitimate request from a service provider is actually an attempt to steal personal data, and this can be done in intelligence gathering for cyber-attacks. Similarly, attacks can appear to be part of a different whole than they really are, as when a social-engineering attack asks a user to briefly change their password to ?test the system? but actually uses that as a loophole to obtain permanent access.

3.7 Speech-act cases

Finally, deception can involve semantic cases related to communication. Most of these have been covered by the previous cases, but it is helpful to distinguish internal and external preconditions. Internal preconditions are on the agent of the action, such as ability of a user to change their password, and external are on the rest of the world such as the ability of a site to accept a particular user-supplied password. Both provide useful deceptions by defenders since it is often hard to confirm deception in such conditions in cyberspace.

3.8 Comparing the taxonomies

Our taxonomy has advantages over the two previously discussed in that it specifies more precisely the deception mechanism, which aids in brainstorming in planning, monitoring of plan execution, and detection of deception. For instance, "mimicking" in the Bell and Whaley taxonomy does not distinguish mimicking the agent (as an attacker pretending to be a system administrator), mimicking the object (as a single honeypot pretending to be thousands of sites (The Honeynet Project, 2004)), or mimicking the cause (as in giving a false error message to an attacker (Rowe, 2004)). Similarly, "camouflage" in the Dunnigan and Nofi taxonomy does not distinguish between camouflaging the mechanism that logs attacker actions (as in the Sebek honeypot software (The Honeynet Project, 2004)), camouflaging the logging site (as in Sebek), or camouflaging the hidden accompaniment to a free download (as in Trojan horses).

4. Examples

To illustrate use of our taxonomy, consider a phishing scam to steal passwords for a later attack.

1) The user receives an email from ?Pay-Pal, Inc.?.
2) The message tells them their account has been compromised and new security measures are being taken to prevent reoccurrence.
3) They are asked to click on a link that says ?Pay-Pal? to go to a Pay-Pal site.
4) On the site, which looks just like the Pay-Pal site, they are asked to enter their account name and password.

Major deceptions are in agent, which is not Pay-Pal, and beneficiary, which is the criminal and not the victim. Another is in the purpose of entering the password, which is to steal it and not to aid security. Others are deception in object and "location-at?, the identity and location of the site that the link takes them to.

Next, consider rootkit installation:

1) An attacker breaks into a site through a buffer overflow (a too-large command argument) on port 225.
2) They add themselves to the list of authorized users to gain permanent administrator access.
3) They replace operating-system files with their own by copying them from their home site.
4) They delete operating-system logs that indicate what they have done.

Here we have deception as to measure and supertype (of the command argument) on port 225. This enables deception in agent by masquerading as a system administrator. This enables them to change parts of the operating system into Trojan horses, which is deception in object, supertype, and accompaniment. Changing the logs is then deception in the cause if other users notice anything unusual.

Here is an example of defensive deceptions for deliberate obstruction of rootkit installation:

1) An attacker breaks into a site through a buffer overflow.
2) The overflow is recognized and their session is secretly transferred to a safer machine.
3) They try to copy files from their home site using FTP, but are told the network is down.
4) They try to copy files using SFTP, but the files are garbled in transit.
5) They successfully send files from their home site using email.
6) When they try to copy the files into the operating-system directories, they get an error message that ?the directory is protected? although it is not.

Here the initial defensive deception is in object and ?location-at? for the site. Then there are two deceptions in external preconditions, one in value, and one in both cause and external precondition.

5. Second-order deceptions

?Second-order? deceptions can be defined as those based on recognition by an agent of one or more of the above ?first-order? deceptions. They primarily involve participant, causal, and speech-act cases, since detection of deception affects perceptions about who participates, why they do it, and the preconditions they recognize. For instance, a defender can attempt rather transparent external-precondition deceptions in an attempt to seem inept, to better fool the attacker with subtler deceptions such those in material and accompaniment as by transferring Trojan horses back to them. Similarly, an attacker can try an obvious denial-of-service attack, a deception in frequency, to camouflage a subtler attack such as a buffer overflow to get administrator privileges, a deception in measure and value. Can there be third-order and higher-order deceptions? Probably not, much in the way that counter-counterdeception is hard to distinguish from plain deception in most analysis.

6. Rating deception methods

As guidance for deception planning, it is helpful to rate the suitability of the methods overall for both offensive and defensive cyberspace deception. We will use a scale of 0 (deception is ineffective) to 10 (deception is highly effective). In addition, some ways of presenting the deceptions will be more convincing than others (Fogg, 2003), an issue analyzed elsewhere (Rowe, 2004).

6.1 Rating offensive deception methods

We rate the threat of offensive deception methods by considering three factors: (1) the counted number of distinct mentions in 314 articles randomly selected from Volume 23 of the Risks Digest (catless.ncl.ac.uk/Risks), a newsletter on new threat types; (2) a report on current trends in cyber-attacks (MessageLabs, 2005); and (3) our personal estimate of the mountability and effectiveness of the deception type based on knowledge of capabilities of software. In following list, the first number in parentheses is our overall assessment of the seriousness of the threat posed by the deception method, and the second number is its count from the Risks Digest sample.

- agent (10, 34): Intrinsic to identity deceptions of many kinds.
- accompaniment (10, 8): Intrinsic to Trojan horses.
- frequency (9, 3): Intrinsic to denial-of-service attacks.
- object (8, 14): This includes camouflaged and decoy files, executables, and other software objects, which are often easy in cyberspace because of the lack of supporting details such as handwriting or provenance (Mintz, 2002). Examples are the email impersonations of legitimate businesses and the fake Web sites used in phishing attacks.
- supertype (7, 5): Disguise an attack as something else, the classic strategy of social engineering.
- experiencer (6, 4): Eavesdrop to steal secrets.
- instrument (6, 3): Attack with surprising software tools.
- whole (6, 3): Do actions that do not appear overall to be suspicious but that include the attack. Important in spyware and other Trojan horses, and also in social engineering.
- content (5, 2): Make it appear that data is something else.
- external precondition (5, 0): Make impossible requests.
- measure (5, 2): Send data too large to handle, as in buffer overflows.
- location-from (4, 4): Attack from a surprising site.
- purpose (4, 4): Give incorrect reasons during social engineering.
- beneficiary (4, 2): Make a victim falsely think they are the beneficiary, as in email scams.
- time-at (4, 0): Attack at a surprising time.
- value (3, 3): Send a different piece of data than expected, as in buffer overflows.
- location-to (3, 2): Attack a surprising destination.
- location-through (3, 0): Use a surprising site as an intermediary.
- time-through (3, 1): Take much more than the usual amount of time to do something.

- internal precondition (2, 2): Pretend inability to do something, or pretend ability to do something bad as a means of extortion.
- direction (2, 0): Attack in the reverse direction to that expected.
- effect (2, 2): Pretend that something did not work properly.
- cause (1, 0): Give false explanations of events.
- location-at (0, 0)
- orientation (0, 0)
- time-from (0, 0)
- time-to (0, 0)
- recipient (0, 0)
- contradiction (0, 0)
- manner (0, 0)
- material (0, 0)
- order (0, 0)

6.2 Rating defensive deception methods

Defense from cyber-attacks provides just as many opportunities for deception but these are less well known. Deceptions can be triggered by reports from an intrusion-detection system (Proctor, 2001) that a suspicious user is present (Monteiro, 2003). Here our ratings for suitability rely more on statistics of observed attack types from the MessageLabs report and from www.securitystats.com, and our own literature survey and analysis of feasibility and effectiveness (as explained in each item) since the Risks Digest had only 10 instances of defensive deception in our sample. Based on the former information, we assume that the major offensive threats in cyber-space are in decreasing order of importance: (1) rootkit installation; (2) viruses and worms; (3) theft of secrets; (4) fraud; (5) sabotage; (6) denial of service; (7) theft of services; (8) site defacement.

- external precondition (10): Give a false excuse as to why you cannot do something that an attacker wants. Excuses are excellent tactics for not doing things (Snyder, Higgins, & Stucky, 1983), and they work well in cyberspace because users are accustomed to frequent cryptic or erroneous error messages (Rowe, 2004).
- effect (9): Lie about what a command did, as in saying a file was downloaded when it was not, or saying that a command failed when it was ignored (Rowe, 2003). Users of computers often accept unexplained failures, as when executables get damaged, so they are surprisingly willing to accept outright lies by software.
- content (9): Plant disinformation in computer files for an attacker to find. Much fake data in computers is easy to construct automatically (Rowe, 2006), although fake prose requires skill (Gerwehr et al, 2000). Doctored images can be powerful disinformation, and are increasingly easy to construct (Hutchinson & Warren, 2001). Deception in content also includes redefining programs to do unexpected things, as in Trojan horses in email attachments that claim useful functions but actually insert viruses and spyware into computers.
- time-through (8): Delay someone you are suspicious about by pretending that a computation is taking a long time. This is especially good against a denial-of service attack since such attacks depend on speed; delays simulate a successful attack while maintaining normal service for other users (Julian, Rowe, & Michael, 2003).
- purpose (8): Pretend to be a different kind of site than you really are. Honeypots do this, and whole networks of decoy military sites could be created using old real data suitably modified. Or ask questions to deliberately delay an attacker or help trace them.
- experiencer (8): Monitor the attacker through hidden logging to a secure remote site that the attacker cannot erase, as with the Sebek tool; or pretend to report events to law enforcement.
- value (7): Systematically misunderstand commands, as by losing the first character in every command line or by encrypting a file when not asked. Or deliberately follow a procedure incorrectly that has been specified by an attacker in a social-engineering attack, such as calling the wrong office to get a new password.
- cause (7): Give a false error message such as an unexpected "segmentation error"; this is often coupled to deception in effect. Good excuses can refer to causes that an attacker does not understand, but it is hard estimate the knowledge of your attacker (Rowe, 2004).
- object (7): Camouflage software objects to confuse the attacker or provide Trojan horses. The Sebek tool conceals itself from the directory-listing program, but concealing some running software is hard, and hiding files that should normally be present is suspicious. Phishing-type attacks involving camouflaged sites can also be used to trick attackers into giving information about themselves. Desirable fake data like false credit-card numbers and passwords (?honeytokens?) can also represent deception in object (Cohen & Koike, 2003).

- frequency (7): Give too much information to an attacker to overwhelm them, as a kind of defensive denial-of-service attack. Or unnecessarily ask for confirmation on every command, or dazzle the attacker with so many fake sites or files that they cannot find the real ones (Cohen, 1999).
- measure (6): Create messages or files that are unexpectedly large, analogously to deception in frequency.
- location-to (6): Deceptively forward attacker commands to a less-dangerous environment ("sandbox"), as is done in "honeypot farms" (The Honeynet Project, 2004).
- internal precondition (6): Give a false excuse for command failures alleging internal problems like overloading or new software (Rowe, 2004). Also, a target of social engineering can pretend to be inept to encourage an attacker to underestimate them.
- supertype (5): Be a decoy (unharmable copy) for important site (like a military command-and-control site).
- agent (4): Run a ?sting? on a phishing scam by pretending to be a foolable consumer, then trace the money; or pretend to be a hacker and chat with other hackers to collect intelligence.
- accompaniment (4): Send a Trojan horse back to the attacker by encouraging them to download software containing it.
- direction (3): Deceptively transfer files with Trojan horses back to the attacker as they send files to you.
- material (3): ?Emulate? (simulate hardware in software) to provide a safe imitation of a target computer, as is done in testing attacks in the DETER testbed (www.deter.org).
- location-from (2): Try to scare an attacker by fake messages from law enforcement.
- time-at (2): Falsify directory time information (Rowe, 1996).
- whole (2): Ask occasional questions of an attacker during chat to help identify them.
- time-from (1): Falsify directory time information.
- time-to (1): Similar to the preceding.
- instrument (1): Do something an unexpected way.
- location-at (0)
- location-through (0)
- orientation (0)
- beneficiary (0)
- recipient (0)
- contradiction (0)
- manner (0)
- order (0)

7. Putting the deceptions together

As an example of a coordinated defensive deception plan, suppose we create a network of honeypots (a ?honeynet?) to fool attackers of a military network. The honeynet could have the names of real command-and-control sites (deception in object, supertype, and "location-to") with real-looking data (deception in object and content). The data could be real data with changed dates and times (deception in "time-from" and "time-to"), referring to false locations (deception in "location-at"), and involving nonexistent people (deception in agent). The system could secretly report all user commands to a secure remote site (deception in experiencer). If the attacker wants to launch an attack from this network, the system could lie that the outgoing network connection is down (deception in external precondition) or is being debugged (deception in internal precondition). When the attacker wants to download files, it could lie that the transfer utility is not working (deception in external precondition); it could just observe that files are not being transferred properly today (deception in effect); it could damage the files in transit (deception in content); or it could delay a long time (deception in "time-through"). To irritate the attacker, it could ask many questions requiring confirmation (deception in frequency) or tell them unnecessary information about processing status (deception in measure). It could also secretly transfer the attacker to a safer ?sandbox? site if the attacker appears to be particularly dangerous (deception in "location-to"), or it secretly send Trojan horses back to the attacker as the attacker downloads files to it (deception in accompaniment and direction).

Putting deceptions together this way has a synergistic effect because they help support one another. Multiple first-order deceptions also provide opportunities for second-order deceptions. For instance, one can be quite obvious during file downloads about delaying by asking unnecessary confirmations to cover the modification of executable files in transit to prevent them from working once installed. This is a second-order deception in internal precondition, as the more obvious deceptions (in "time-through", frequency, and external preconditions) make it appear that the defender is inept.

8. Conclusions

Deceptions in cyberspace can cover a wide range of techniques, and it is important to be familiar with all of them in military planning. We have presented a taxonomy more fine-grained than any previously advanced, and it should be useful for planning. But there is a separate issue we have not addressed here of evaluating the effectiveness of deceptions in context, as by principles (Fowler & Nesbitt, 1999) or by mathematical metrics (Rowe, 2004; Rowe, 2006). Will increased use of deception increase adversary deception in return? Probably, but the escalation cannot continue indefinitely because as deceptions become more common they become less effective, and as they become more complex to maintain effectiveness they become harder to plan and maintain. This will mainly help the defender since defensive deceptions can generally be simpler and easier to create. The situation may be analogous to that of computer viruses, which are now a lesser threat as the increased frequency and sophistication of antiviral software is forcing attackers to work harder.

9. Acknowledgements

This work was supported by the National Science Foundation under the Cyber Trust Program. The views expressed are those of the author and do not represent policy of the U.S. Government.

10. References

Austin, J. L. (1975) *How to Do Things with Words, 2nd edition*, ed. Urmson, J., & Sbis, M., Oxford University Press, Oxford.

Bell, J. B., & Whaley, B. (1991) *Cheating and Deception*, Transaction Publishers, New Brunswick, New Jersey.

Chirillo, J. (2002) *Hack Attacks Revealed*, Wiley, New York.

Cohen, F., (1999) "Simulating cyber attacks, defenses, and consequences," [online] all.net/journal/ntb/simulate/simulate.html.

Cohen, F., and Koike, D. (2003) "Leading attackers through attack graphs with deceptions," *Computers and Security*, Vol. 22, no. 5, pp. 402-411, 2003.

Copeck, T., Delisle, S., & Szparkowicz, S. (1992). Parsing and case interpretation in TANKA. Conference on Computational Linguistics, Nantes, France, pp. 1008-1023.

De Rosis, F., Castelfranchi, C., Carofiglio, V., & Grassano, R. (2003) "Can computers deliberately deceive? A simulation tool and its application to Turing?s imitation game," *Computational Intelligence*, Vol. 19, No. 3, 2003, pp. 235-263.

Dunnigan, J. F., & Nofi, A. A. (2001) *Victory and Deceit, 2nd edition: Deception and Trickery in War*, Writers Press Books, San Jose, California.

Fillmore, C. (1968) "The case for case," in *Universals in Linguistic Theory*, ed. Bach, E. & Harns, R., Holt, Rinehart, & Winston, New York.

Fogg, B. (2003) *Persuasive Technology: Using Computers to Change What We Think and Do,* Morgan Kaufman, San Francisco, CA.

Fowler, C. A., & Nesbit, R. F. (1995) "Tactical deception in air-land warfare," *Journal of Electronic Defense*, Vol. 18, No. 6 (June), pp. 37-44 & 76-79.

Gerwehr, S., Weissler, R., Medby, J. J., Anderson, R. H., & Rothenberg, J. (2000) "Employing deception in information systems to thwart adversary reconnaissance-phase activities," Project Memorandum, National Defense Research Institute, Rand Corp., PM-1124-NSA, November.

The Honeynet Project (2004) *Know Your Enemy, Second Edition*, Addison-Wesley, Boston, Massachusetts.

Hutchinson, B., & Warren, M. (2001) *Information Warfare: Corporate Attack and Defence in a Digital World.* Oxford, UK: Butterworth Heinemann.

Julian, D., Rowe, N., & Michael, J. B. (2003) "Experiments with deceptive software responses to buffer-based attacks," IEEE-SMC Workshop on Information Assurance, West Point, New York, June, pp. 43-44.

Latimer, J. (2001) *Deception in War*, The Overlook Press, New York.

MessageLabs (2005) "2005 Annual Security Report," [online], www.messagelabs.com.

Mitnick, K. (2002) *The Art of Deception*, Cyber Age Books, New York.

Mintz, A. P. (ed.) (2002) *Web of Deception: Misinformation on the Internet*, CyberAge Books, New York.

Monteiro, V. (2003) "How intrusion detection can improve software decoy applications," M.S. thesis, U.S. Naval Postgraduate School, [online] www.cs.nps.navy.mil/people/faculty/rowe/oldstudents/monteiro_thesis.htm.

Nyberg, D. (1993) *The Varnished Truth: Truth Telling and Deceiving in Ordinary Life,* University of Chicago Press,

Chicago.
Proctor, P. E. (2001) *Practical Intrusion Detection Handbook*, Prentice-Hall PTR, Upper Saddle River, NJ.
Rowe, N. C. (2003) "Counterplanning deceptions to foil cyber-attack plans," IEEE-SMC Workshop on Information Assurance, West Point, NY, June, pp. 203-211.
Rowe, N. (2004) "Designing good deceptions in defense of information systems," *Computer Security Applications Conference*, Tucson, AZ, December, pp. 418-427.
Rowe, N. (2006) "Measuring the effectiveness of honeypot counter-counterdeception," Hawaii International Conference on Systems Sciences, Poipu, HI, January.
Rowe, N., & Rothstein, H. (2004) "Two taxonomies of deception for attacks on information systems," *Journal of Information Warfare, 3* (2), July, pp. 27-39.
Snyder, C. R., Higgins, R. L., and Stucky, R. J. (1983) *Excuses: Masquerades in Search of Grace*, Wiley, New York.