MALICE THROUGH THE LOOKING GLASS: BEHAVIOUR ANALYSIS FOR THE NEXT DECADE

Jeff Debrosse, David Harley
ESET Research, 610 West Ash Street, Suite 1900, San Diego, CA 92101, USA

Email {jdebrosse, dharley}@eset.com

ABSTRACT

Most VB attendees have a major interest in malicious code. Often they focus on the highly technical issues around the intricacies of malware technology and counter-technology, the programmatic detail of attack and counter-attack. Sometimes they focus instead on the higher level application of defensive technology to corporate or infrastructural environments, even the entire Internet. More rarely, they look at the human side of malware management, mostly from the point of view of involving the potential victim (individual or organization under attack) in the defensive process (education and training, policy enforcement and so on).

However, malware is only part of a complex process of malicious exploitation. Behaviour analysis is a crucial topic in 21st century anti-malware, but rather than focusing purely on programmatic behaviour, should we not be looking at the psychosocial behaviours that underpin the exploitation mechanism? (By this we mean not only the behaviour of the criminal, but that of the victim.) This paper considers steps towards a holistic approach to behaviour analysis that would enable us to treat the disease rather than the symptom, drawing on both social and computer science.

INTRODUCTION

The anti-malware industry is still largely focused on code analysis, not on social engineering [1]. We do attempt (well, some of us do) to educate and inoculate against malicious social engineering by instilling scepticism [2], but we don't do our own social engineering in a programmatic context. Where we parse textual content, we're looking for specific code-related, system-related or network-related material like DNS data or links to malicious code or traffic, or indications of the presence of malicious code such as registry entries.

I'M GONNA BE A (SOCIAL) ENGINEER [3]

What, then, do we mean by social engineering? Clearly, we're not using it in the limited sense in which we're accustomed to seeing the term used in the context of malware – where it's usually applied to the use of verbal trickery and deception in order to persuade individuals to execute malware – though that's certainly relevant to our thesis. Nor are we referring specifically to the theft of passwords by psychological subversion, though that's also relevant. In fact, the term attracts a wide range of definitions [4]. In the security industry it's almost de rigueur to focus on the security threats that come under the 'social engineering' umbrella (indeed, one of the authors has been criticized for insisting on using a more academic, legally/morally/ethically neutral definition in the security context [5]. It isn't surprising: the security industry (and especially its malware-focused sector) has historically been conservative, technology-focused and essentially reactive. We've learned over the years that we can't survive in the current threatscape by clinging to the Gods and Ants 'we know what you need better than you do' culture that predominated in the 1990s. In other words, we've moved on from some of the right-wing dogmata of the past:

- Supplying purely reactive signature-based technology
- Cherry-picking our targets ('This is antivirus software, and that's not a virus!')
- 'Disinfection is a second-class option: replace or reinstall!'
- 'If you have to ask how, you aren't qualified!'
- 'Sharing samples? Wash your mouth out!'
- '97% of users are idiots.' [6] (OK, that one still has some adherents.)

While some of us may miss the simplicity of those days, increased flexibility makes us more useful to our customers, even if those customers, the media, and even other sectors of the security industry continue to see us as marginalized, self-isolating and autocratic dwellers in ivory towers [7].

TECHNOLOGY VERSUS SOCIOLOGY

Nonetheless (and in this respect we're no different from most of the security industry), we continue to focus on technical solutions, at any rate in a development context. While a significant proportion of us believe that education does 'work' [2], in practice, we largely work on that approach through informational white papers, blogs, vendor-neutral conference papers and articles, community projects (either in the anti-malware community with initiatives like AMTSO [8] or outside in the 'real' world [9]) and so on. This is important work (well, we would say that, wouldn't we?), and comprises a significant proportion of what we mean by 'social engineering', but it it's far from all-inclusive.

Time for some definitions: these come from an earlier FAQ on the topic [10].

- Deceptive practices that attempt to obtain information from people using social, business or technical discourse [11].
- A method of 'sounding' information which is not generally accessible. Often, perpetrators will pose as insiders by using pertinent keywords during conversations and thus receive information useful for other purposes [12].
- Term used among crackers and samurai for cracking techniques that rely on weaknesses in wetware rather than software; the aim is to trick people into revealing passwords or other information that compromises a target system's security [13].

Clearly, these all come from the security-focused side of the fence. In fact, the paper by Goudey [5] includes an excellent pragmatic definition that covers most of these areas generically (including the informal malware-focused definition coined earlier):

'Social engineering techniques are used to target the human interface (as opposed to hardware and/or software) in order to

compromise data and/or systems [for the attacker's benefit]. These techniques are comprised of a two-step approach that initially deceives targets to perceive a false reality, and then takes advantage of particular social psychological traits to convince targets to act on the misinformation contained in the message.'

PSYCHOLOGY AND DECEPTION

This makes the case clearly for social engineering as an instance of the study of deception in social psychology – in Goudey's case, using Computer Mediated Communication (CMC). However, it doesn't meet our needs. As in the paper Goudey was quoting [4], we want to return to a more traditional and somewhat academic, ethically neutral definition closer to the term's roots in social sciences in order to 'accentuate the positive' [14]. Let's look at a couple of definitions from that paper:

- Hidano [15] defines the purpose of social engineering as to resolve social problems by 'social recognition and measurement method, integrated theory of Psychology, Sociology and Economics, spatial and social design theory, and people's participation and decision forum.'
- Jacobs [16] uses the definition 'the discipline and quantitative constraints of engineering....applied to social legislation.'

These are nearer the older meaning, but for our purposes this covers all the bases: 'Psychological manipulation of an individual or set of individuals to produce a desired effect on their behaviour.' [4]

For the purposes of this paper, our desired effect is clearly to increase the security of our customers. While we'd all like to think that we're doing the best we can to enhance our technologies programmatically using such malware identification technologies as advanced heuristics, in-the-cloud approaches, sandboxing, generics and so on, our focus in terms of software development is still on program identification – mostly, we look for:

- Known bad objects (blacklisting)
- Known good objects (whitelisting, change detection)
- Presumed bad objects (generic blocking, detection by class, filetype etc.)
- Presumed good objects (still whitelisting and change detection)

THE PSYCHOLOGY OF DETECTION

Let's think for a minute about detection. We've already indicated that scanning software detects by code analysis, which is true in a literal sense. When we talk about behavioural analysis in this industry, we're usually talking about programmatic behaviour. If we can't identify known badware/goodware (software whose behaviour is already known and classified), we assign the object we're detecting to one class or other by analysing its behaviour. We can do this either predictively by static analysis or dynamically, by seeing what it does in an emulated or virtualized environment. At this point, we may identify it as (for instance):

- Something already classified as goodware or badware.
- Something resembling known badware closely enough to be classified as badware in its own right.
- Something that shares characteristics associated with badware, even though it doesn't resemble a known badware family.
- Something which may be good or bad, but has proscribed characteristics.
- Something below the 'suspicious' threshold.

Clearly, some of these pre-existing classifications have been through a lab-based analytical process that involves making a good/bad value judgement. In the past, when anti-malware was almost entirely focused on viruses, this was easy. As Orwell might have said, 'Replicative bad, non-replicative good'. Unless, of course, you're thinking about trojans, in which case you're faced with a broader, more conventional class of malice. But that's OK, because while you can't formally detect malicious intent programmatically, you can identify programmatic characteristics that are commonly associated with malicious intent.

Of course, it's not really that easy: there are many instances where the borders are somewhat blurred (Q: 'Is it a remote access tool or is it a backdoor trojan?' A: 'That depends on who's using it.') Even worse, there are instances where software that is clearly meant to be malicious, deceptive, fraudulent etc. is presented as legitimate not only to the intended victim, but to the legal system, where producers of adware, spyware, rogue anti-virus and so on may be prepared to defend the 'legitimacy' of their product in court [17]. Indeed, they may assert that the detection of such products is malicious, monopolistic, and so on, pointing to minimal or purely imaginary 'security features', their own advertising claims, and low-value or deceptive reviews and certifications as 'proof' of legitimacy while chipping away at what little trust the world at large has in legitimate security software [7].

The current escalation of malicious programs masquerading as security programs confirms that we have long passed the point where it's feasible to rely on automated lab analysis to establish the legitimacy of (some) malware. Despite the immense overhead that manual analysis and motivational second-guessing can entail, the risks inherent in getting it wrong are, potentially, uncomfortably expensive. But given this already existing (and increasing) overhead, is it really a good idea to start looking at user behaviour as well? And what does behaviour analysis have to do with social engineering?

THE HUMAN ELEMENT

It is often said that the weakest link in the security chain is the 'human element'[18]: (hence, perhaps, the use of near-pejorative terms such as luser, meatware, or wetware). Systems can be hardened to many varying degrees of complexity and security, but all the technology and procedures cannot adequately protect the end-user from his own ignorance. Time and again, methodologies such as phishing have been proven to be quite effective at aiding the cybercriminal in his quest to harvest credentials or other valuable data from the end-user.

The challenge of applying myriad technologies to protect the end-users from the almost limitless spectrum of fraud campaigns is, quite literally, never-ending. The criminal element, motivated by the potential for financial gain, has waged a non-stop psychological attack aimed squarely at the most vulnerable layer in relation to Internet usage. While

many of us are either vaguely or extremely familiar with the seven-layer OSI (Open Systems Interconnect) stack, too many security experts are insufficiently well-versed with what can be considered the eighth layer – the end-user. This paper is focused on that eighth layer and how to more adequately protect the end-user from the vulnerabilities inherent in his own, human nature.

On the other hand, crime is a problem that has not been solved in many thousands of years of recorded history. Criminals typically attempt to circumvent controls as well as to take advantage of people – both physically and psychologically. When we refer to exploiting the human element, this attack comes from the psychological vector – preying on well-understood [4] human needs, wants and desires to gain personal and/or private information through deceit and manipulation. 'Social engineering' in this security-oriented sense is of such concern that SANS lists it in the 'Top 20' security risks [19].

John Maynard Keynes said, 'It is better to be roughly right than precisely wrong.' [20] Expecting the eighth layer to protect itself against the onslaught of attacks by becoming more aware of the complex permutations of security solutions to protect against technological and psychological attacks does not scale, and is precisely wrong. Of the two types of attack, psychological attacks via social engineering can be said to be the most difficult to protect against due to the complexity of human actions and reactions. The flip side to that statement is that there are various actions that, for the most part, will be repeated time and again by an overwhelmingly large number of end-users – thus guaranteeing the attacker some return on their investment. 'If there's one thing sure thing, when it comes to security, it's that people make the same mistakes – over and over and over. It's something that hackers have come to count on.' [21] Kevin Mitnick is an example of someone who knew that regardless of the technological security implementations companies deployed, it was overwhelmingly simple social engineering (albeit often in combination with technical knowledge) that granted him access to the information that he was targeting [22].

The current global population numbers at approximately 6.7 billion people [23] with the Internet penetration rate at 24% – this comes out to a staggering 1.6 billion Internet users (and potential victims) globally [24]. When considering that the eighth layer is over 1.6 billion strong and growing, it is to be expected that the criminal element will regard the ever-growing number of Internet users as rich and fertile ground for illicit revenue-generating opportunities through psychological profiling, targeting and manipulation.

THE PEOPLE PROBLEM

Attacking the human layer is accomplished through social engineering via the following methods, among others:

- Phishing and other messaging scams
- Phone
- Face-to-face
- Scareware and ransomware
- Website/URL trust establishment

For the security community, saving the end-user from their own ignorance would be a panacea indeed. Unfortunately, due to the endless combinations of motivations, emotions, goals, desires, and so on, gauging the human response to various stimuli and from multiple modes (multi-modal) is exceptionally difficult at best. It is this particular difficulty that makes attacking the human layer an omnipresent threat. While we can create signatures for viruses, rules for packet filtering and even highly accurate heuristic algorithms to protect against malware and network-borne threats, we have yet to design a system or methodology to reduce the risk to the end-user by leveraging a human behaviour-predictive algorithm or any risk mitigation technique that evaluates the end-user and protects them accordingly.

PROBABILITY AND EMAIL

'The most important questions of life are, for the most part, really only problems of probability.' [25]

Probability theory is a branch of mathematics that deals with the analysis of large data sets of random phenomena. Various statistical patterns can be inferred by simply watching a series of actions and outcomes over a period of time. With more time and data an observer can begin to see patterns emerging and will be able to determine the outcome of future events based on actions that they have observed. Leveraging their success/failure ratio will allow them to make more accurate predictions for future results.

Probability theory is currently used in many systems that defend end-users from the onslaught of spam. A method widely employed is called Bayesian [1] spam filtering, which is based on Bayes' Theorem – a theorem of probability theory originally stated by the Reverend Thomas Bayes. The specific method is called a 'Naïve Bayes Classifier' and is used when there are high-count dimensional data sets to evaluate. A naïve Bayes classifier works well primarily because it measures performance using a 0-1 loss function which counts the number of incorrect classifications rather than a measure of how accurate the classifier estimates the posterior probabilities [26]. Two of the key benefits of using naïve Bayes classifiers are that the method incurs low computational overhead and provides for very fast machine learning [27].

Basically, if a person wants to determine the probability of a particular outcome, one can look at a sampling of information from previous (related) events and come to a fairly accurate determination of the outcome.

When using Bayesian analysis to detect spam, a subset of email is tested for different conditions and the probability of a word, or combinations of words, causing a particular email message to be classified as spam is determined. This has proven to be significantly more effective against spam than content filtering. Bayesian analysis learns from previous email (both spam and non-spam email). The end result is a highly adaptive machine learning process that has resulted in increasingly accurate detection and classification of spam. The following are typical criteria used to determine the probability that an email message is spam:

- The body of the message
- Various combinations of words
- Metadata – including HTML (i.e. colours, various tags, etc.)

The following is an example of Bayesian spam filtering in action:

1. An email is scanned for various data that would classify it as spam.
2. The phrase 'male enhancement' is detected in the body of the email (85% probability of the message being spam).
3. The subject contains the phrase 'real prescription meds' (95% probability).
4. The body also contains the word (FREE) in all caps (98% probability).
5. The sender's email address and sending server are different – and the server is not authorized to send email on behalf of the domain name in the sender's address (100% probability).

Bayesian analysis is only one of the analytical tools available for the application of information theory to text analysis, of course: for instance, Markov chains make use of the probability that certain tokens will follow other tokens [1, 28].

PROBABILITY AND PEOPLE

Taking Bayesian analysis to the next level, predicting the outcome of human behaviour based on sampling of past behaviour can be viewed as a very difficult task – and indeed it is. For instance, in the case of network security, when a person receives a link in an email or via a website, depending on the end-user's emotional state or surroundings, they may take action and click on a particularly dangerous link to go to a website that they would not normally visit. The challenge here is in learning the particular user's web browsing habits and making a probabilistic determination of their next move – and protecting them from their own actions.

Today, behavioural targeting is doing something very similar. Behavioural targeting works by monitoring and tracking sites that users have visited as well as the content they have viewed. This process has typically been performed by browser 'cookies' – small files that store, for later retrieval, information regarding a user's use of a particular website. This information is used to repeatedly analyse the user's behaviour and to create a profile (model) of that particular user's wants and needs. This allows a website owner to predict the likes, dislikes and *actions* of the users visiting that particular site and assign them to a particular demographic. 'The results of this behavioural targeting proved to be quite dramatic. When compared with the basic web ads, the behaviour ads were seen by 115 per cent more business travellers making at least one trip a year. The targeted consumers also scored three per cent higher than the average viewers in brand awareness.' [29]

People are targeted in many ways by advertisers and cybercriminals. For instance, there is a portion of the human brain called the reticular activator. The Reticular Activating System (RAS) is generally considered to be a prime factor in arousal and motivation: however, it's commonly believed that the reticular activator mechanism has a filtering function that has a very direct effect on marketing, in terms of targeting and susceptibility. Quite simply, it determines what information the brain receives at any particular time – as well as what to ignore. Spear phishers and those that craft spam email also (unknowingly, in most cases) can be regarded as targeting the reticular activator system. For instance, consider a pop singer who is popular worldwide, you can quickly predict the amount of fake video links, spam and other material that will be created to leverage this popular person's fan base, which could number in the millions. Now that there is an awareness of this pop star by their fans, every news story, email and website mentioning that particular pop star's name will get the fan's attention. It happens because, as with any filtering techniques, there is now a 'notification rule' in the brain that alerts the person to any mention of their favourite pop star. During times of heightened awareness of entertainers and celebrities, disasters or other emotionally moving events, the cybercriminal is clearly attempting to take advantage of the new filter rule placed in the brain by the reticular activator system. (You could also consider this mechanism at a more abstract level as conscious exploitation of a memeplex [30], but perhaps we should resist that particular branch-line on this occasion.)

GET YOUR GAME (THEORY) ON

The research field of game theory is generally considered to have been created by John von Neumann and Oskar Morgenstern with the publication of their book, *Theory of Games and Economic Behavior* (1944). Game theory is a form of mathematics that attempts to predict behaviour in any sort of 'strategic' environment – from simple interaction between two people to the movement of financial markets, and even modern-day warfare. A classic example of game theory is portrayed in what is known as 'The Prisoner's Dilemma'. Below is a typical scenario and set of outcomes for this particular problem in game theory:

Two burglars, B1 and B2, are captured near the scene of a burglary and are questioned in separate rooms. Each has to choose whether or not to confess and implicate the other. Here is the matrix of options each prisoner is facing:

- If neither burglar confesses, then both will serve one year on a charge of carrying a concealed weapon.
- If each confesses and implicates the other, both will go to prison for 10 years.
- If one burglar confesses and implicates the other, and the other burglar does not confess, the one who has collaborated with the police will go free, while the other burglar will go to prison for 20 years.

The strategies are: confess or don't confess. The payoffs are: the sentences served – which is expressed in the below 'payoff table'.

		A1	
		confess	**don't**
B2	**confess**	10,10	0,20
	don't	20,0	1,1

Here's how the table works:

- Each prisoner chooses one of the two strategies
- A1 chooses a column and B1 chooses a row
- The two numbers in each cell tell the outcomes for the two prisoners when the corresponding pair of strategies is chosen.
- The number to the left of the comma tells the payoff to the person who chooses the rows (B1) while the number to the right of the comma tells the payoff to the person who chooses the columns (A1). Therefore (reading down

the first column) if they both confess, each gets 10 years, but if A1 confesses and B1 does not, B1 gets 20 and A1 goes free.

A1 might reason as follows: 'Two things can happen: B1 can confess or B1 can keep quiet. Suppose B1 confesses. Then I get 20 years if I don't confess, 10 years if I do, so in that case it's best to confess. On the other hand, if B1 doesn't confess, and I don't either, I get a year; but in that case, if I confess I can go free. Either way, it's best if I confess. Therefore, I'll confess.'

B1 can, and presumably will, reason in the same way – so that they both confess and go to prison for 10 years each. Yet, if they had acted 'irrationally', and kept quiet, they each could have gotten off with one year each.

While there are many variations of this problem and many other examples of game theory in action, the Prisoner's Dilemma is, quite possibly, the best-known example. Below is another example of the application of game theory – being used in physical security [31].

'A research project from the University of Southern California (USC) developed randomization security software. It is called ARMOR (Assistant for Randomized Monitoring of Routes).

'Here's how it works: Computer software records the locations of routine, random vehicle checkpoints and canine searches at the airport. Police then provide data on possible terrorist targets and their relative importance. These data may change from one day to the next, or if there have been any security breaches or suspicious activity.

'The computer runs, and – voilà – police get a model of where to go, and when. The software comes up with random decisions that are based on calculated probabilities of a terrorist attack at those locations, using mathematical algorithms.

'The result: Security with airtight unpredictability. With the software, it's extremely difficult to predict police operations.'

CONCLUSION

It is clear that one of the final frontiers of network security is the human element. What has not been so clear is how to protect this weak link in the security chain effectively and directly.

After much research, it has become clear that taking game theory to the next level – determining the most likely action that a user will take in a given situation, enabling the reinforcement of 'safe' decisions and the sanctioning (or at least monitoring) of 'unsafe' decisions – can make for a much more secure computing environment for the end-user because their security software would be able to more accurately determine the outcome of their actions. This protection could be bolstered by leveraging the concepts behind behavioural targeting, being aware of the relevant theory behind psychological and physiological systems, behind the reticular activation system, and applying probability theory and Bayesian analysis to an individual's behaviour/interaction with their computing environment. The conclusion of this research is that a great, untapped opportunity awaits. That opportunity is the creation and effective implementation of a human behaviour heuristic for the end-user.

How would such a heuristic system be implemented and used? Well, that's quite a conceptual leap, and a number of questions would need to be answered.

- What data would it process in order to monitor behaviour? Would it monitor web interactions, parse log files and messaging content?
- What feedback mechanism would be most effective? While mild electric shocks would probably not be acceptable to most end-users, there is plenty of evidence that 'are you sure?' pop-ups, nag screens and prompts for administrator passwords have limited success and inspire not only user-resistance, but creativity in finding evasion strategies.
- Are we still talking about anti-malware applications at all? After all, the main thrust in anti-malware technology these days is to reduce footprint and resource usage: adding another layer of functionality, however expertly engineered, is unlikely to result in a *reduction* of system impact.
- What are the ethical implications of software that monitors user behaviour? Would it need to be restricted to an interaction between the user and the local application? What would be the implications of a centrally managed console application somewhere in the IT centre?
- Would the feedback mechanism be more effective if it threw (presumably anonymized) data back into the cloud to be aggregated with other behavioural data for use in improving product performance at the interface between security and ergonomics? There's certainly a case to consider by analogy with the way that some products now pass back threat data for aggregation and consequent tuning of detection algorithms and technologies.

These are not trivial questions. However, given the potential benefits of moving some of the armouring mechanism from the technology to the end-user, they are surely worth trying to answer.

REFERENCES

[1] Harley, D., Abrams R. Whatever Happened to the Unlikely Lads? Proceedings of the 19th Virus Bulletin International Conference, 2009. http://www.eset.com/download/whitepapers/People_Patching.pdf.

[2] Abrams, R., Harley, D. People Patching: is user education of any use at all? Proceedings of the AVAR Conference, 2008.

[3] Seeger, P. I'm gonna be an engineer. http://www.mudcat.org/@displaysong.cfm?SongID=2980.

[4] Harley, D. Raising the Titanic. Proceedings of the EICAR Conference, 1998.

[5] Goudey, H. Exploiting vulnerabilities in the human interface for fun and profit. Virus Bulletin, August 2008. http://www.virusbtn.com/spambulletin/archive/2005/08/sb200508-exploiting-vulnerabilities.

[6] http://www.yuikee.com.hk/info-ctr/newsletter/ykcl-news01-12.pdf.

[7] Harley, D. I'm OK, you're not OK. Virus Bulletin, November 2006. http://www.virusbtn.com/virusbulletin/archive/2006/11/vb200611-OK.

[8] http://www.amtso.org/.

[9] Abrams, R. Securing our eCity. http://www.eset.com/threat-center/blog/?p=1077.

[10] Harley, D. Social Engineering FAQ, 2001. (Currently unobtainable: withdrawn from website pending update.)

[11] SRI International, quoted in Cobb, S. The NCSA Guide to PC and LAN Security. McGraw-Hill, 1996.

[12] IT Baseline Protection Manual. Bundesamt für Sicherheit in der Informationstechnik, 1996.

[13] The Jargon File. http://catb.org/jargon/html/S/social-engineering.html.

[14] http://en.wikipedia.org/wiki/Ac-Cent-Tchu-Ate_the_Positive.

[15] Hidano, N. Social Engineering. URL valid at May 1997 but no longer available.

[16] Jacobs, J. Why 'Social Engineering' is an Oxymoron. URL valid at May 1997 but no longer available.

[17] Malcho, J. Is there a lawyer in the Lab? Proceedings of the 19th Virus Bulletin International Conference, 2009.

[18] Gutmann, P. Bugs in the Wetware? The Psychology of Computer Insecurity. http://www.cs.auckland.ac.nz/~pgut001/pubs/psychology.pdf.

[19] http://www.sans.org/top20/.

[20] As quoted by Canestrelli, E. Current Topics in Quantitative Finance, 1999.

[21] Vaas, L. Inside the mind of a hacker. eWeek, July 2009.

[22] Mitnick, K., Simon, W. The Art of Deception. Wiley, 2002.

[23 http://www.census.gov/.

[24] http://internetworldstats.com/stats.htm.

[25] de Laplace, P.S. Theorie Analytique des Probabilities.

[26] Cowan, R. Predictive Modeling and the Bayes Classifier. http://www.cs.umd.edu/~samir/498/rosa.ppt.

[27] Stern, M.K.; Beck, J.E.; Park Woolf, B. Naïve Bayes Classifiers for User Modeling.

[28] Overton, M. Canning more than SPAM with Bayesian Filtering. Procedings of the 14th Virus Bulletin International Conference, 2004.

[29] Baker, L. Behavioral Targeting and Contextual Advertising. http://www.searchenginejournal.com/behavioral-targeting-and-contextual-advertising/836/.

[30] Blackmore, S. The Meme Machine. Oxford University Press, 1999.

[31] Talwalkar, P. Game Theory for Airport Security? It's Happening: ARMOR at LAX. Mind Your Decisions. 30 April 2008. http://mindyourdecisions.com/blog/2008/04/30/game-theory-will-keep-me-safe-at-the-airport-armor-and-lax/.