FEATURE

You are the Weakest Link, Goodbye! – Malware Social Engineering Comes of Age

Martin Overton,
ChekWARE UK

The Year of the Snake in the Chinese astrological calendar ran from 24 January 2001 to 12 February 2002. In Western culture snakes are seen as treacherous, sly, sneaky and the very embodiment of evil. They are credited as being able to 'hypnotize' or 'charm' their prey, so that they appear entranced and pliable to the will of the snake (this has no apparent basis in fact, as the intended prey is probably just frozen by fear).

Does this sound familiar in the realms of malware, with the virus authors, or more specifically the social engineering aspect that has become prevalent in malware recently 'hypnotizing' their victims? Certainly, it would seem that 2001 could be dubbed the Year of the Malware Social Engineer.

More Social than Most?

It has been obvious for some time that many virus writers have been watching and learning from the hackers (I use the term in the way most 'non-anoraks' use the word – those real 'old-time hackers' out there, please forgive me for using the media definition).

The appearance of W32/MyParty@MM and W32/Porman@MM (not to mention W32/Nimda@MM) has shown that scenarios (nightmare or not) discussed by researchers behind closed doors have also been thought of by those in the wider world – specifically those in the malware-writing world (much to the vexation of the virus researchers and those of us in large organizations, who are forced to play 'chicken' with the latest and greatest of the 'just-released' threats).

Several months ago, I pondered with a few other security professionals why virus writers hadn't taken the next obvious route and used Web (http-like) addresses (as attachments to the email, rather than a URL in the body of the message which links to a Web site) as a means of increasing the likelihood of their creations being executed by the less vigilant of their victim pool.

The obvious attachment names to use, from a social engineering point of view, are Web addresses ending in '.com' (minus the http:// prefix), as most computers will happily run such a misnamed file as an executable – especially if it is a binary file format (i.e. *.COM, *.EXE).

The use of social engineering is not new in the world of malware; however, I believe it has now 'come of age' (just coming up to the teenager stage, in fact, and bringing all its problems, attitude and teenage angst with it).

At the very least, social engineering is the current flavour-of-the-year for 'suckering' potential victims, by using 'new' ways to tempt them into taking the bait.

Luckily, some of the victim pool have wised up to the fact that email attachments (from strangers) may spell danger, and treat them as the electronic equivalent of 'Typhoid Mary'. However, that is not usually the case when the unexpected files are received from someone they know.

Yet again, we have seen malware authors 'steal' another recommended way to deal with the thorny issue of email attachments. Many companies (both AV vendors and IT Security departments) have suggested that directing recipients to a URL instead of sending a file attachment is 'safer'. Time to think again, maybe?

What is Social Engineering?

Here's a short, but very apt, definition from Jargon File: *'**Social engineering n.** Term used among crackers and samurai (hackers for hire) for techniques that rely on weaknesses in wetware (people) rather than hardware or software.'* (See http://www.tuxedo.org/~esr/jargon/html/entry/social-engineering.html.)

In his book *Secrets and Lies*, Bruce Schneier lists social engineering as one of six 'aspects of the human problem' when focusing on information systems security. He states that social engineering is 'very effective', and that it goes straight to the 'weakest link in any security system: the poor human being trying to get his job done, and wanting to help out if he [or she] can.' (See http://www.rr.sans.org/securitybasics/awareness.php.)

In reality, however, social engineering is lots of things and it is even harder to pin-down when it is used in relation to malware, but the key to it all is the following: 'Someone wants something you have (or have access to) or wants you to perform an action (such as disclose information, run a program). To achieve this, the would-be Social Engineer will lie (claim to be someone or something they are not, or that they have access to something they are not entitled to), cheat (forge credentials or get you to run code that does something to escalate rights or install a backdoor by convincing you that it is something else) and steal (data, passwords, identities, availability of system resources)'.

In short, the would-be Social Engineer plays on the natural human tendency to trust, and to want to help others.

Damned if we do, Damned if we don't

Part of the problem is the fact that 'we' (that is a collective we, not a royal one) have anti-virus solutions in place in many companies: at the mail gateway, on file and print servers, internal mail servers, http and ftp gateways, and on our final bastion, the desktop.

This 'multi-layered-approach' actually appears to exacerbate the problem, as end users seem to be more willing to take a risk with an attachment or link because they know that the company (and therefore their computer) is 'protected' by anti-virus software. This leads them into an 'it-can't-happen-to-me' attitude, and even if it does happen, the users tend to see it as being not their problem, but an IT problem.

Sex, Lies and Topical Themes

Let us have a look at the some of the various methods and themes that have been tried, and how successful they have been:

Sex:	W32/Pops@MM, W32/Toget@MM, W97M/Melissa@MM, VBS/Loveletter@MM, VBS/VBSWG@MM (Anna Kournikova)
Fear:	W32/Whitebait@MM
Greed:	Nigerian Money Transfer and its many variants[1]
Altruism:	W32/SirCam@MM, W32/Myparty@MM, SULFNBK.EXE Hoax
Authority:	Appears to come from someone you know or trust (such as the almighty MS or an Anti-Virus/Security company)
Humour:	W97M/Comical@MM, W32/Roach@MM
Games/ ScreenSavers:	W32/Maldal.d@MM, BudFrogs Hoax
Topical:	W32/Ska@M (Happy99), W32/Maldal.c@MM
Anti-virus or Security updates:	W32/Whitebait@MM
Double extensions:	W32/Magistr@MM, W32/Sircam@MM, W32/Badtrans@MM
'Polymorphism':	W32/SirCam@MM, W32/Gokar@MM, W32/Klez@MM

('Polymorphism' in this instance refers not to true polymorphism, but to random subject lines, body text, attachment names and extensions.)

Well, how successful were some of those?

Very successful (either very fast-burners or had a long-term presence): W97M/Melissa@MM, W32/Sircam@MM, VBS/Loveletter@MM, W32/Badtrans@MM, W32/Hybris@MM, W32/Magistr@MM, VBS/VBSWG@MM.

Quite successful (either fast-burners or had a long-ish presence): W32/Goner@MM, W32/Maldal.d@MM, W32/Ska@m, W32/Maldal.c@MM, SULFNBK, BudFrogs Hoax.

Average success (never really took off): W32/Klez.

Poor (a mere drop in the ocean): W32/Whitebait@MM, W32/Roach@MM.

The Scores on the Doors

Why did some of those listed above do far better (i.e. spread further and/or more rapidly) than others?

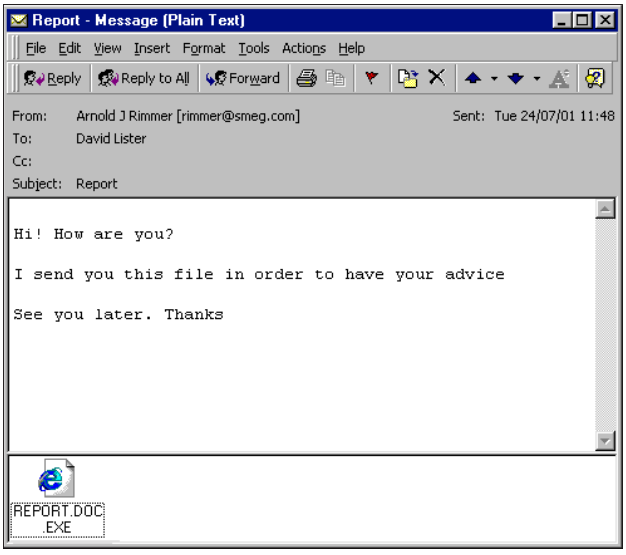
Friend or foe? a major plus point for successful distribution of malware seems to be the arrival of the file from someone the intended victim *knows*. This appears to be the key factor for 'Electronic Ephemera' (hoaxes and their kin).

Psychological buttons: those pieces of malware that use the typical 'high-interest psychological buttons' seemed to do best – sex, greed, altruism and topical themes.

Timing: this can make a big difference, but it plays a small part in the overall 'malware-that-got-lucky' model.

Holes: those pieces of malware using known exploits in target operating systems or applications worked very well. This is a trend I expect to be taken to new levels this year, making W32/Nimda@MM seem like a minor annoyance.

Originality: those pieces of malware that are significantly different from any other (the first of their ilk) tend to be more successful, when combined with other factors listed above, as they are less likely to be detected either heuristically or via generic (family) signatures.



Own SMTP engine: this seems almost to be the *de facto* standard for today's mass-mailers, and tends to improve the ability for a mass-mailer to spread.

The Solutions

Let's have a look at the possible solutions to some of the problems which social engineering malware adds to the 'overall' malware problem in an organization:

i) Technology

Can the problem be solved completely purely by throwing technology at it?

Of course not, you'd need a full integrity management system (to identify/block changes) as well as a change management system (to check whether any changes were authorized changes) and an expert system to attempt to manage the whole process. Even with all of these precautions in place there is no guarantee that an unauthorized change couldn't happen.

Even if we succeeded in building a better snake trap, the snake would evolve so that it could avoid or bypass the trap. It quickly develops into a snake and mongoose (malware writer and anti-virus/security company) game, and the victims (end users and their companies, aka the 'mice') are the innocent casualties of the game (i.e. 'lunch').

If technology is the answer, then why have many companies suffered with the likes of W32/Nimda@MM, W32/Goner@MM and many other mass-mailers (many of which have inbuilt social engineering elements)? Obviously if technology is the 'whole' answer, then something is not working.

ii) Education

I am on record as stating that trying to educate end users about malware is mainly a waste of time and I stand by that still, even though I made the statement as long ago as 1996 when I was a virgin *VB* Conference presenter.

In large companies educating the end users is like painting a very long bridge: once you've reached the end of the bridge, it is time to start painting from the beginning again.

Furthermore, trying to educate end users about malware issues is like expecting a car owner to understand how the car works, when all they really need to know is how to use it, when to fill it up, check the tyres, send it in for repair (or get it towed), and what to do when things go wrong (who to call).

In other words, a simple check list is all the end users need. They don't need to know how to strip the engine, nor understand the mechanics or physics involved.

To take this metaphor further, your support staff can be likened to garage mechanics; *they* are worth training. This is also true with regard to your in-house developers and systems administrators and other operations staff.

Finally, educating end users in large companies is very expensive, both in terms of the cost of hiring suitably qualified trainers and in the lost productivity of the trainees.

In the light of this, many companies see the occasional outbreak as an acceptable risk that they are (currently) prepared to accept and sign off.

iii) Policy and Procedures

So what will work with your end-users? Give them simple guidelines, policies and procedures for them to follow, which are easy to understand (and which do not consist of lots of 'techie-speak').

Below are some simple guidelines for end users that could help in combating the threat from social engineering-based malware:

- Follow 'Safe Hex' guidelines, which should be available at http://yourcompanyintranet.com/safehex/. (Most AV companies have some basic guidelines which you can adapt for your own company. Otherwise see: 'Safe Hex in the 21st Century', *VB* June 2000 p.16 and *VB* July 2000 p.14 for some suggestions and guidance.)
- Send all received warning emails, or suspect files to 'suspect@yourcompany.com' – a central email drop-box for your company which is monitored by a team (or member of staff) that understands malware and related issues and knows where and how to verify or debunk received files or warnings.

For your support and operations staff:

- Create an intranet site, and put policy, procedure and FAQs, virus warnings, hoax information and other pertinent documents there.
- Publicize a 'hot-line' number and encourage employees to use it.
- Join Security Alert Mailing Lists.
- Monitor AV/security/hoax Web sites.
- Join an Early Warning System-type service, such as AVIEN's EWS (see http://www.avien.org/).
- Ensure your systems are patched.
- Roll-out anti-virus updates as soon as you can (after testing them of course).
- Ensure that you educate your users via a simple, easy-to-understand security policy, which is underpinned with good and well-documented processes and procedures.
- Make it clear to staff that not following the guidelines, etc. could lead to disciplinary action, and it may even cost them their job.

As with most things in life, a mix of approaches or a 'holistic' approach offers the most appropriate way of dealing with malware threats, including social engineering malware. Use whichever of the suggestions listed in this article will work in your environment. There are almost certainly others that I haven't covered here.

Conclusions

It seems clear (to me) that the use of social engineering in malware is likely to increase for the rest of the year and I believe it will be the most effective way for malware authors to ensure that their creations achieve a wide and receptive audience.

The current increasing 'trend' of mass-mailing malware can be likened to rape. Statistics show that most rapes (71 percent in the US and as many as 97 percent in the UK) are committed by someone the victim knows and probably trusts (see http://abc.eznettools.net/D302506/X329849/stats.html and http:/www.rapecrisis.co.uk/statistics.htm). In the case of mass-mailing malware, both the sender and recipient are, more often than not, both 'victims'.

We need to ensure that end-users are more careful with all electronic information that they receive from those they know and trust, as well as from strangers.

We need to establish in our user-base the need to be more cautious, suspicious and a little more paranoid, to help minimize the chances of them becoming part of the problem, rather than part of the solution. If we fail to do this, then we too are also part of the problem, and not part of the solution (which should be our goal).

However, you must remember that (in most cases) it is ultimately a human being pushing the buttons (and having their psychological buttons pushed) and this is the root of the problem.

The 'key' to breaking this cycle, I believe, is to make the end users accountable, make security their problem too, and remove the focus from it being an IT (a technology) issue, to what it (social engineering) is – a 'human problem'.

The old maxim states: *'Curiosity killed the cat'*. How many lives do your end users have left? In fact, the character Fox Mulder from the *X-Files* TV series may have the ultimate mantra for our staff to learn … 'Trust No One!'

Footnote

[1] There are many reports from both the UK and the USA that a surprising number of mugs … er, I mean unsuspecting victims lost a significant amount of money, and occasionally their lives as a result of being taken in by the 'Nigerian Money Transfer'. So much so, in fact, that it has been subject to both an FBI fraud alert (see their Web site http://newyork.fbi.gov/contact/fo/nyfo/fraudalert.htm) and a warning from the US Secret Service (see http://www.ustreas.gov/usss/index.htm?alert419.htm).