

BEHAVIOURAL MODELLING OF SOCIAL ENGINEERING-BASED MALICIOUS SOFTWARE

Matthew Braverman

Microsoft Corporation, 1 Microsoft Way,
Redmond, WA 98052, USA

Tel +1 425 703 2229

Email mattbrav@microsoft.com

ABSTRACT

Some of the most active threats in the wild today exploit weaknesses in the component with the largest attack surface area in the end-to-end operation of a computer: the user. Malicious software such as Sober, Netsky, Bagle and Mywife can take control of a computer not because of any software bug or vulnerability, but because they somehow lure the user to execute them, usually by running an attachment of an email.

This paper will provide examples of poignant social engineering 'exploits' over the past few years and attempt to construct a model, using telemetry from *Microsoft's Windows Malicious Software Removal Tool*, that can predict the prevalence of a specific social engineering threat based on its characteristics and appeal to the user.

INTRODUCTION

In June 2006, *Microsoft* released a report detailing trends gathered by the release of the *Windows Malicious Software Removal Tool (MSRT)* from January 2005 to March 2006. During this period, the tool was executed 2.7 billion times and removed malicious software from 5.7 million unique computers. Of the 5.7 million machines cleaned by the tool, 35% were infected by some malicious software capable of infecting a computer only by using social engineering [1]. This is a significant figure because it illustrates the prevalence of malware that leverages social engineering and clarifies how the malware landscape is far from restricted to malicious software that exploits vulnerabilities in software.

This paper will focus specifically on examining malware that leverages social engineering to infect a computer, where social engineering is defined as 'a non-technical kind of intrusion that relies heavily on human interaction and often involves tricking other people to break normal security procedures' [2]. It will review techniques used both in the past and present and will use up-to-date data (as of the writing of this report) from the *MSRT* to differentiate those social engineering techniques which have been particularly successful.

INFECTION TYPE BACKGROUND

Various data shows that, each year, millions of computers become infected with malicious software (or 'malware'). A common question asked by security researchers and corporate IT administrators alike is 'how was the malware able to infect the computer?'. The answer to this question is key to being able to create and establish appropriate security measures aimed at preventing a similar attack in the future. The 'how' usually fits into one of four categories, described below.

- *Remotely exploitable software vulnerability*. In this case, the malware is able to infect a computer by exploiting a remotely exploitable vulnerability in some software running on the computer. Note that, in this category, infection through the existence of the vulnerability is effectively completely 'automatic' and requires no manual interaction on the part of the end user.
- *Remotely exploitable policy vulnerability*. As in the above case, malware that is able to infect a machine by exploiting a remotely exploitable policy vulnerability does so without requiring any action on the part of the end user. However, in this category, the malware does not exploit a vulnerability or bug in software but instead, some weak configuration, policy, or access control setting on the computer. The most common example of this technique is malware that infects a computer by brute-forcing a weak password on that computer.
- *Social engineering*. In many cases, to infect a computer malware resorts to a variety of techniques to somehow encourage the user to take some action which would allow the malware to execute. In most cases, this requires the user to visit some website (which may then exploit some vulnerability to run the malware) or run some program.
- *Non-social-engineering-based manual execution*. While manual execution of malicious software can usually be traced to some form of social engineering, in some cases infection can occur simply as part of a user's normal computing routine. For example, consider a website which exploits a vulnerability such that, when a user visits the site, the site infects his/her computer automatically with malicious software. Also consider that the malware sends email to users enticing them to visit the site. Users who reach the site by falling victim to the encouragement of the email are the victims of social engineering. However, users who just happen to wander to the site by typing in the URL as part of their daily web-surfing are unfortunate victims of their own routine.

As malware has evolved to resemble more 'blended threats', it has become difficult to determine exactly which of the above techniques were used to infect a computer because some malicious programs use several of them to attempt to infiltrate a machine. For example, bots are usually capable of attempting to infect a computer using a remotely exploitable software vulnerability as well as attempting to crack a weak password on a target computer. Determining which of these techniques enabled the malware to infect the computer is difficult with current anti-malware offerings, however this is surely an area for technological improvement in the future. Similarly, in some cases, determining the difference between manual infection by social engineering and non-social-engineering-based methods is difficult.

As noted in the introduction, this paper will concentrate on social engineering threats but will, at times, refer to some of the other categories listed above for comparison.

SOCIAL ENGINEERING TECHNIQUES: PAST, PRESENT AND FUTURE

The concept of malicious software leveraging social engineering to infect a computer is far from new. However, it

is a practice that has become significantly more commonplace in the last decade with the increased popularity of the Internet and subsequent malicious software which somehow uses that medium to replicate. Realistically, the boot sector and DOS-based file infector viruses that first popularized malicious software on PCs in the mid-1980s to mid-1990s relied more on manual infection through non-social-engineering-based techniques. For example, most users who managed to infect their computer with a boot sector virus did so because they booted their computer with a floppy disk from a friend; however, the malware itself was not responsible for engaging a user to use the disk. Similarly, file-infector viruses required a user to initiate a transfer of an infected file to another user and then the second user to execute that file, likely unaware that it was infected. Whereas modern malicious programs have the capability to distribute and advertise themselves over such media as email and instant messaging applications, file-infector viruses could not, for example, post themselves to a dial-up bulletin board system (BBS).

This section will review popular social engineering techniques that have appeared over the past decade. Later in the paper, the categories introduced in this section will be used to determine the success of certain characteristics in attracting the user's attention.

The two basic components of a social engineering threat are an object (e.g. a message) which somehow triggers the attack and a delivery vehicle for transporting the object. This paper will enumerate and discuss the three most common delivery vehicles for transporting such objects:

- Email
- Live chat (including instant messaging and Internet Relay Chat or 'IRC')
- File-sharing networks (accessible through peer-to-peer applications)

This paper will also enumerate various social engineering techniques based on how an attacker manipulates an object. For the purposes of this discussion, we will assume that objects contain, at most, three basic parts:

- Message content (e.g. the message body/subject)
- A sender
- A reference to the malicious software (e.g. an attachment or a link)

Note that, depending on the delivery vehicle, all of these components may not be present. Most notably, in the case of most peer-to-peer applications, only the reference to the malicious software is available. No sender information or accompanying message is present in this case.

DELIVERY VEHICLES

Email

With email taking hold as one of the first key 'killer Internet apps' for a wide spectrum of Internet users, it was somewhat expected that virus authors would take advantage of its collaborative and social capabilities to spread malicious software. While there are threats which spread over email using mechanisms besides social engineering, in the context of this paper, this section will focus specifically on social engineering malware.

The most basic type of social engineering email worm features a subject, message body, and/or attachment name that encourages the end user to take action, where action could be executing the infected attachment of the email or visiting a website which may install malicious software on the user's computer. One of the first and most infamous examples of this technique appeared in March 1999, with Melissa, a *Word* macro virus (and then shortly thereafter with LoveLetter in 2000). While *Word* macro viruses were hardly unique at that point in time, Melissa was unique in its usage of email (and specifically *Microsoft Outlook*) to replicate. The virus sent copies of itself to users stored within the infected user's address book with a subject of 'Important Message From <infected user's name>' and a message body of 'Here is that document you asked for ... dont show anyone else ;-)'.

Live chat applications

While email is an extremely powerful form of communication, it is limited by application and protocol-level overheads associated with composing a message, adding recipients and sending that message. Thus, this medium is not conducive to scenarios in which two (or more) individuals want to collaborate or engage in a real-time conversation. This is the gap that live chat applications have sought to fill. While such scenarios have long been possible between computer enthusiasts (via custom applications and the Internet Relay Chat), one of the first mainstream implementations of a live chat application can likely be traced to AOL's *Instant Message* feature as part of its online service.

It wasn't until the late 1990s that software vendors began to offer mainstream live chat applications as standalone, free applications. AOL also pioneered this path with its *AOL Instant Messenger* application. However, it was quickly followed by such popular instant messaging clients as *Microsoft's MSN Messenger*, *Yahoo!'s Yahoo! Messenger*, and *Trillian's ICQ* client. Today, there are a large number of niche instant messaging applications. Many of these applications are targeted at and popular with specific audiences (e.g. a specific geographical region or high-security corporate/military environments).

Thus, as with email, live chat/instant messaging applications soon became subject to malicious code using these mechanisms to replicate. At this point, all mainstream instant messaging programs have had malicious code written specifically to leverage their clients to propagate. In most of the cases, the malware is programmed to correspond to a specific program, but some are capable of replicating over several programs.

Note that some malicious software (e.g. bots) use live chat applications (especially IRC) as a communications mechanism to communicate between some sort of server and a set of infected clients or zombies. While some vendors classify these threats as instant messaging worms, this paper will recognize instant messaging worms only as those that use the mechanism to *replicate*.

File-sharing networks

File-sharing networks first gained significant popularity around the turn of the millennium/century, mainly buoyed by the MP3 music-sharing craze of that time. While exchanging files through various methods (including the two other

delivery vehicles described above) was certainly feasible, they were limited by the fact that they required some other individual to initiate the transfer. File-sharing applications operate on a peer-to-peer basis and assemble files from a large collection of users into one central point which other users can request at their leisure. In most cases, each application manages its own central database; there are few overlaps or examples of sharing between the applications.

In the past few years, however, malicious software has polluted the collection of files offered by specific file-sharing applications by sharing themselves as part of the collection alongside and disguised as 'legitimate' files. The word 'legitimate' is enclosed in quotation marks since a common technique for the malicious software is to pose as the software application hacks and cracks regularly sought out by users of these file-sharing applications, which are hardly legitimate. Malicious programs that exhibit this behaviour are usually referred to as P2P (peer-to-peer) worms.

Malicious programs that leverage file-sharing networks to spread are slightly different from those which leverage email or live chat applications, since file-sharing networks offer the user only a filename and not a corresponding message or sender. Therefore, the main and only target of social engineering is the filename. We consider this social engineering malware because the filename provides significant context to the user which entices them to run the file – this is validated by the presence of significant prevalent malware that uses this technique.

MESSAGE CONTENT

The most basic (and sometimes most effective) manifestation of a social engineering attack is through the content of the message. Regardless of how dangerous a link or attachment looks, if the attacker can come up with message text that entices the user or piques his/her interest into opening the attachment, that attack succeeds.

A fair amount of research has been done by anti-malware experts on the types of human characteristics and emotions that social engineering threats attempt to duplicate with message bodies. One of the most recent and thorough examinations, 'The Signs, Signifiers and Semiotics of the Successful Semantic Attack' was performed by Myles Jordan (an employee of CA at the time, now at *Microsoft*) and Heather Goudey (CA) and presented at EICAR 2005 [3]. The paper reviews a taxonomy of 12 social psychological vulnerabilities (inexperience, curiosity, greed, diffidence, courtesy, self-love, credulity, desire, lust, dread, reciprocity, friendliness) commonly used by 20 of the most successful Internet worms from 2001 to 2004. The objective of this paper is not to repeat this research but to build on it by offering specific examples that highlight these psychological vulnerabilities and to provide data that evaluates their effectiveness.

Listed below are several popular types of message bodies which malicious software has leveraged.

Generic conversation

This technique is one of the most basic, but at the same time most effective, because it appeals to the curiosity and friendliness of the victim. It usually involves text designed to appear to be a continuation of a previous (possibly,

off-Internet) conversation and/or something that the receiver could possibly expect. By keeping the text as short and generic as possible, the attacker increases the odds that it could possibly trigger some sense of familiarity with the reader.

In the case of emails, many examples in this category include content only in the Subject field of the message (e.g. 'Hi' or 'Hello'). In cases where a message body exists, the content is usually broad and generic (e.g. 'Here's that file we were talking about').

Non-English language used

Most malicious software in circulation today uses English as its language of choice, including as part of disseminating a social engineering attack. In some cases, English is clearly not the first language of the malware author; however, given the pervasiveness of the language, malware authors usually prefer to write in English that may be difficult to comprehend rather than use a non-English language.

That said, some malware in the wild today uses messages in which the content is in a language other than English. There are at least two reasons for this approach:

- First, the author might simply be trying to confuse the reader or make him/her curious about the content, even if he/she is not able to read the language used.
- Second, the more likely reason is that this is a threat aimed at a specific geographic region.

Virus alert/software patch required

In some cases, virus authors have tried to capitalize on the recent successes of their own creations by composing messages that pose as virus or software vulnerability alerts and include a 'fix' to protect the computer from the virus and/or vulnerability. In many cases, these messages are engineered to appear to originate from a legitimate source, such as *Microsoft* or another well known anti-virus provider.

Malware found on your computer

A variant of the above case, in this technique, the message will imply that a specific virus was found on the victim's computer and ask him/her to execute an attached fix to remove the threat.

No malware found

Some legitimate email server anti-virus scanners will add a footer to each email message scanned, indicating that the mail was scanned and found to be clean (or that malicious software was found and removed). Some malware today will insert a similar/authentic-looking footer at the bottom of the mails that they distribute in an attempt to raise the reader's confidence that the mail is clean.

Account information

One common technique used by social engineering malware is to formulate a message that poses as an alert or informational message from an administrator or a customer service department pertaining to an issue (e.g. the user's password needs to be reset or the account has expired) with an account supposedly owned by the victim. The message usually indicates that some action needs to be taken with

respect to this account (involving, for example, opening a file or visiting a website).

Mail delivery error

Most email users, at one time or another, have received a legitimate Non-Deliverable Receipt (NDR) message when an address they sent an email to turned out not to exist. Some malware will attempt to pose as NDR messages, with the malicious attachment purporting to be the attached message.

Physical attraction

Some of the most common social engineering attacks apply to the physical curiosities of their victims, offering glimpses of explicit pictures or videos. The most infamous example of this technique is the Anna Kournikova script virus which promised readers pictures of the tennis star, sans clothing.

Accusatory

Another common theme in messages distributed by social engineering malware is that the user has done something wrong and needs to take action to rectify the situation (e.g. visited an illegal or improper website or distributed malware). These messages usually purport to arrive from some figure of authority (usually a law enforcement organization or possibly an administrator of an account owned by the user) and prompt the user to take further action to rectify the situation (which will result in them infecting their computer).

Current events

Some messages sent by malware rely on the user's interest in issues and events currently popular in the world. Major sports events and global/national tragedies such as Hurricane Katrina, the events of September 11, etc. are prime examples.

Free stuff

While somewhat similar to the 'physical attraction' category, this class of message attempts to entice the victim into thinking that he or she will receive some sort of positive incentive by opening the message and running the attachment. In most of these cases, the incentive takes the form of a program capable of cracking software copy or licensing protection or even the commercial software itself.

SENDER

A message is made even more relevant and personal when the attacker is able to manipulate it so that it appears to have been sent from another individual that the victim trusts, or at least knows. An attacker can accomplish this either by implicitly impersonating another user, or by doing so explicitly.

Implicit sender spoofing

When an attacker is able to infect a victim's computer with malicious software, the malicious software can use (e.g. programmatically) the legitimate software on the victim's computer to send messages directly to other users. Thus, the messages would appear to come from the victim, not the attacker. In this paper, this is referred to as implicit spoofing because the attacker does not have to take action to manipulate the message.

This technique was very common with early email viruses such as Melissa. The virus would infect a computer and then use the programming interface of the victim's email application (e.g. *Microsoft Outlook*) to send mail. Some of these applications have since evolved to help prevent such interaction. However, this practice is still very common with instant messaging worms.

Explicit sender spoofing

Explicit sender spoofing takes place when the attacker creates and forges a message to look as if the message came from someone possibly known to the victim. In this case, the attacker does not necessarily have to infect this person's computer to spoof the message.

For example, this technique has become especially common with email worms after applications such as *Microsoft Outlook* helped prevent most implicit sender spoofing scenarios. In most cases, the worm will infect a victim's computer, search it for email addresses, and then use its own embedded outgoing SMTP email client to craft a message using one of those email addresses as the sender. This technique also leads to confusion if the victim's computer detects the virus, as the victim will warn the sender that his/her machine is infected with a virus, which would not be the case.

ATTACHMENTS

Extension alterations

The LoveLetter virus was one of the first email viruses to use the social engineering technique of double file extensions, a practice still common today. With this technique, the malware names the infected attachment using the schema:

```
<name>.<fake, innocuous-looking extension>.<real, executable extension>
```

For example, ILOVEYOU.txt.vbs. In this case, the right-most extension is the 'real' one that *Windows* will use to launch the associated file handler. However, the objective of the technique is for the user to mistake the fake extension (e.g. 'txt' in our example) as the real one and regard the file as harmless.

Since LoveLetter, this technique has seen some evolution by other malware. For example, some malware will insert a significant number of spaces between the fake extension and the real extension. This will result in some email clients hiding the real extension from view.

Icon manipulation

Some malware will modify their program icon to mimic an icon that is usually identified as being the symbol for a file handler for usually innocuous files. For example, a notebook is the icon for the *Windows Notepad* program, typically used to open text files. By using that icon (and the above technique), users are lulled further into a false sense of trustworthiness by a very effective visual trick.

Attachment relevance

Worms such as Magistr and Sircam (appearing in early and mid-2001 respectively) implemented other social engineering-oriented attachment infection schemes designed

to make the user more confident about opening the infected file attached to the message.

Magistr searches for documents on the infected user's hard drive and uses the contents of those documents to build the subject and body of the message it sends. While, in some cases, this technique can result in a jumbled mess of incomprehensible content for the recipient, in other cases, it can be so highly relevant that the recipient would not hesitate to open the attachment. Magistr may even attach harmless copies of these documents to the email (in addition to the virus) to throw the recipient off the scent. Finally, to distribute itself, Magistr infects an existing file and distributes that as the attachment instead of sending out an exact copy of itself, again as an attempt at feigning relevance to the recipient.

Sircam used a similar technique, by attaching itself to a document stored on the infected user's hard drive, adding an executable attachment to the file, and then using that new file as the attachment to the message. When the user runs the attachment, the malware takes control and then runs the original file, giving the user the illusion that the attachment is legitimate and clean.

Attachment archiving

With the introduction of attachment blocking in *Microsoft Outlook* for specific executable attachments, malicious software has evolved to find new ways to transmit malicious executable code to users. One of the most common methods in use today is to leverage a technique commonly used by individuals to get legitimate executables past this protection mechanism: packaging the malicious code within an archive (usually a zip or rar file).

A further evolution of this technique both packages the malicious code within an archive and password-protects the archive, including the password within the text of the email for the user to enter. The objective of this method is more to bypass anti-virus scanners (most of which are capable of scanning within zip and rar files) than to entice users to open the attachment. In some cases, this technique hinders execution, since users have to take the additional steps of reading the password from the mail and entering it (and, in the case of rar files, usually installing a rar decompression program). However, some threats have been fairly effective in crafting their messages to appear legitimate, describing how entering the password is necessary to get the legitimate attachment past the company's email scanners.

Post-execution social engineering

The main objective of social engineering as implemented by malicious software is to trick the user into *executing* the software. However, some malicious software takes this technique a step further by trying to convince the user that what he/she just ran was indeed legitimate. This is a useful and effective technique especially for fooling experienced users who may be suspicious if the program they just ran provides no interface or information back to the user. If, however, the program shows an error or status message (both popular implementations of this technique) that appears legitimate and seems to match the pre-execution social engineering,

even experienced users may be unable to recognize anything out of the ordinary.

PATTERNS AND TRENDS IN SOCIAL ENGINEERING ATTACKS

Now that we have described a set of behaviours commonly used in social engineering attacks, we will use data collected by *Microsoft* deliverables to determine which of these behaviours are in use today. Specifically, this report will leverage data from the *Windows Malicious Software Removal Tool*, current as of the writing of this report.

This section is divided into three sub-sections with each sub-section going into greater detail on social-engineering related trends.

- In the first sub-section, we will examine social engineering threats at a macro level, measuring how prevalent they are compared to some of the other types of threats discussed in the introduction.
- Next, we will focus on the top social engineering threats detected by the tool and examine which of the above characteristics the threats exhibit.
- Finally, we will examine a specific malware family with a reasonable amount of similarity between the variants to contrast the prevalence of the individual variants with the social engineering characteristics each exhibits.

Note that the malware-naming convention used in the following sections are from *Microsoft's* virus encyclopedia. *Microsoft's* virus encyclopedia was also the primary reference for characteristic information reflected below. However, to help ensure accuracy and consistency, the virus encyclopedias of *McAfee* [4], *Symantec* [5], *Trend Micro* [6], and *CA* [7] were accessed as well. Credit is also owed to the *VGrep* [8] system which was used to resolve naming conflicts.

PREVALENCE OF SOCIAL ENGINEERING THREATS

Figure 1 displays the various malicious software types targeted by the *MSRT* and the number of computers from which each threat type has been removed. Note that threat types are not mutually exclusive, meaning that a piece of malware can fit into several different types. The first three types shown on the graph (mass-mailing worm, P2P worm, and instant messaging worm) comprise the malware types that are associated with social engineering. In total, at least

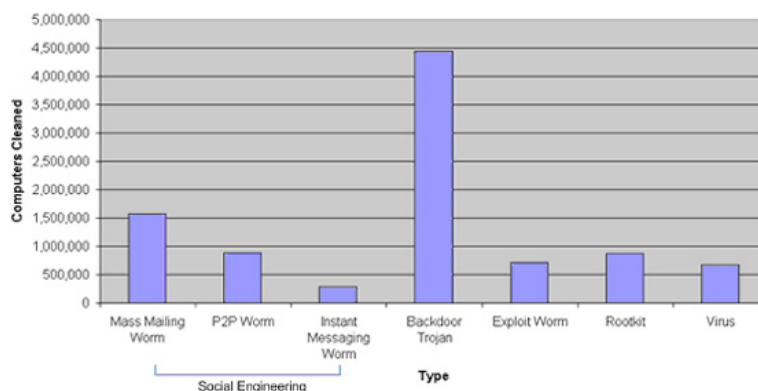


Figure 1: Malware types removed by the *MSRT*.

2.5 million of the 7.5 million computers cleaned were infected with a form of malicious software that leveraged social engineering.

The graph clearly shows mass-mailing worms as the most common type of social engineering malware and the second most common type overall, after backdoor trojans. This is somewhat expected given the ubiquity of email and thus its attractiveness as a vector for spreading malicious code.

What’s interesting, however, is the high number of computers cleaned of P2P worms, especially compared to the low number of instant messaging worms. This is a trend that was first discussed in the *MSRT* report. This data would seem to indicate that replicating malware through peer-to-peer networks is more pervasive compared to instant messaging or live chat applications. Likely one of the main reasons for this difference is the security mechanisms that several instant messaging applications have added to their programs in recent versions, including integrated virus scanning and attachment blocking by extension.

While users should certainly be vigilant about instant messaging threats, they should also be aware of the meaning of statistics presented by various security vendors. If one categorizes bots which communicate through IRC, but do not replicate through IRC, as instant messaging worms, it obscures the true prevalence of worms that do use instant messaging to propagate.

TOP SOCIAL ENGINEERING THREATS

Figure 2 shows the top 20 malicious software variants which leverage social engineering to infect a computer. The chart also shows:

- The rank of the variant relative to *all* malware removed by the tool. In other words, the top 20 variants that leverage social engineering appear in the top 77 of all variants, representing approximately 25% of those top 77.
- The unique number of computers infected by the malware, as identified by the *MSRT*.
- The initial discovery date (month/year) of the malware.
- The date that detection for the malware variant was added to the *MSRT*. This date and the discovery date are included to attempt to reveal any obvious biases that these variables could introduce into the data (e.g. older

malware being more prevalent independent of distribution).

- The social-engineering-specific distribution technique(s) used by the malware variant.

The following observations can be extracted from this figure:

- The most common method of distribution of these top threats is email, with 14 of the 20 variants.
- The top variant (Alcan.B) only uses file sharing to propagate, demonstrating the effectiveness of this distribution technique.
- None of the top 20 social engineering threats replicate through live chat applications.
- Only three of the variants listed, all belonging to the Netsky family, leverage both email and file-sharing applications to propagate.
- There is no clear correlation between the discovery date of a malware variant and its prevalence. The top 20 is composed of a mix of malware discovered as recently as March 2006 and as long ago as August 2003.
- Similarly, there is no clear correlation to the date that detection for a variant was added to *MSRT*, the present day, and its prevalence.

Figure 3 displays the top 20 social engineering threats in the same order as presented in Figure 2. However, in this figure, we compare the message content characteristics leveraged by each of the threats. In the figure, the characteristics are ordered from left to right in decreasing order of presence among the threats.

By far the most common technique used by prevalent social engineering malware is generic conversation. This is not an unexpected result given people’s tendency towards curiosity, and given that the language used in these mails is often effectively designed to trigger a response based on some previous interaction with the person from whom the attacker has disguised the mail to have been sent.

The fact that the free stuff technique is similarly prevalent is also not surprising, especially considering that many of these cases, per Figure 2, are applicable to peer-to-peer worms which are used for exchanging non-infected free stuff.

The prevalence of the foreign language category is reasonably interesting given that most mainstream malware uses English

Rank	Overall Rank	Malware Name	Machines Cleaned	Discovered	Added to MSRT	Distribution		
						Email	Live Chat	File Sharing
1	1	Worm:Win32/Alcan.B	504,194	Apr-05	Feb-06			■
2	4	Win32/Wukill.F@mm	276,683	Sep-05	Oct-05	■		
3	11	Win32/Netsky.P@mm	119,870	Mar-04	Feb-05	■		■
4	12	Win32/Mywife.E@mm CME-24	155,918	Jan-06	Feb-06	■		
5	18	Win32/Wukill.G@mm	80,082	Oct-05	Nov-05	■		
6	19	Win32/Lovgate.V@mm	77,953	Nov-04	Jun-05	■		
7	25	Worm:Win32/Antinny.A	68,400	Aug-03	Oct-05			■
8	28	Win32/Wukill.J@mm	60,076	Nov-05	Dec-05	■		
9	30	Win32/Lovgate.W@mm	56,213	May-05	Jun-05	■		
10	35	Worm:Win32/Antinny.AH	49,335	Sep-05	Oct-05			■
11	45	Win32/Mabutu.A@mm	33,175	Jul-04	Nov-05	■		
12	46	Win32/Mytob.NJ@mm	32,063	Mar-06	Mar-06	■		
13	48	Worm:Win32/Antinny.AI	29,560	Sep-05	Oct-05			■
14	53	Worm:Win32/Antinny.D	28,184	Apr-05	Oct-05			■
15	55	Win32/Netsky.Z@mm	27,466	Apr-04	Feb-05	■		
16	58	Win32/Sober.Z@mm CME-681	26,651	Nov-05	Dec-05	■		
17	61	Win32/Netsky.D@mm	23,575	Mar-04	Feb-05	■		
18	63	Worm:Win32/Antinny.AV	22,054	Jan-06	Feb-06			■
19	69	Win32/Netsky.C@mm	20,759	Feb-04	Feb-05	■		■
20	77	Win32/Netsky.B@mm	18,604	Feb-04	Feb-05	■		■

Figure 2: Top 20 social engineering malicious software removed by the MSRT.

Malware Name	Generic Conversation	Free Stuff	Foreign Language	Physical Attraction	Mail Error	No Malware Found	Account Information	Malware Found on Computer	Accusatory	Current Events	Virus Alert / Patch Required
Worm:Win32/Alcan.B		■									
Win32/Wukill.F@mm	■		■								
Win32/Netsky.P@mm	■	■		■	■	■	■	■	■		
Win32/Mywife.E@mm!CME-24	■			■							
Win32/Wukill.G@mm	■		■								
Win32/Lovgate.V@mm	■	■			■						
Worm:Win32/Antinny.A	■	■	■								
Win32/Wukill.J@mm	■										
Win32/Lovgate.W@mm	■			■						■	
Worm:Win32/Antinny.AH	■	■	■								
Win32/Mabutu.A@mm	■			■							
Win32/Mytob.NJ@mm						■	■				
Worm:Win32/Antinny.AI		■	■								
Worm:Win32/Antinny.D		■	■								
Win32/Netsky.Z@mm	■										
Win32/Sober.Z@mm!CME-681	■		■	■	■		■		■		
Win32/Netsky.D@mm	■										
Worm:Win32/Antinny.AV	■	■	■								
Win32/Netsky.C@mm	■			■	■	■					
Win32/Netsky.B@mm	■	■		■							

Figure 3: Message content characteristics of top 20 social engineering malicious software removed by the MSRT.

as its distribution language. The two main malware families responsible for the high prevalence are Antinny and Wukill. The former targets Japanese language users and the latter targets Chinese language users. MSRT prevalence data indicates that, in both cases, the majority of those infected were using versions of Windows that reflected the language used by the malware. This fact, combined with the high ranking of these threats, reflects the prevalence of regional threats amongst malware today.

The positioning of malware that leverages physical attraction techniques is mildly surprising. Given the amount of attention such threats receive, it was believed by the author that this technique would be one of the most (if not the most) common. Similarly interesting is how rare the usage of current events (another popular target of news media) is among the prevalent malware listed.

Whereas Figure 3 focuses specifically on the characteristics of the message content, Figure 4 compares the overall characteristics of the messages leveraged by the top 20 social engineering malware. Compared to Figure 3, which shows a wide variety of usage of the different content styles, Figure 4 shows most of the techniques being similarly prevalent. The fact that attachment archiving, sender spoofing, extension alterations and icon manipulation are all prevalent, is expected given that they have all been used in the wild successfully for a long period of time. Many of the variants appear to use post-execution social engineering. Some of the specific techniques leveraged by the malware listed above include:

- Worm:Win32/Alcan.B: poses as a fake installation program and displays an error message.
- Win32/Mywife.E@mm!CME-24: launches an empty zip file with the same name to make the user think he/she launched an actual archive.
- Win32/Sober.Z@mm!CME-681: uses a fake error message.

FAMILY-SPECIFIC RESEARCH

The reality is that many different factors are responsible for the successful or unsuccessful propagation of malicious software. These factors can include:

1. In the case of such threats, the social engineering characteristics described in this paper.
2. The length of time it takes mainstream anti-virus vendors to produce a signature from the initial infection.
3. Initial infection seeding. Several studies have shown that malware spreads the fastest/farthest when a large initial set of users become infected first [9].
4. The replication characteristics of the malware (e.g. how often it sends mail, to whom, etc.).
5. Timing/sheer luck. In some cases, there may be ‘good’ or ‘bad’ times to release malicious software. For example, if a new worm is released around the time of a large power blackout, its replication capabilities will be

Malware Name	Attachment Archiving	Sender Spoofing	Extension Alterations	Post-Execution Social Engineering	Icon Manipulation	Replies Unread Messages	Attachment Relevancy
Worm:Win32/Alcan.B	■			■			
Win32/Wukill.F@mm		■		■	■		
Win32/Netsky.P@mm	■		■	■	■		
Win32/Mywife.E@mm!CME-24			■	■	■		
Win32/Wukill.G@mm		■		■			
Win32/Lovgate.V@mm	■	■				■	
Worm:Win32/Antinny.A	■		■	■	■		
Win32/Wukill.J@mm		■		■			
Win32/Lovgate.W@mm		■				■	
Worm:Win32/Antinny.AH	■		■	■	■		
Win32/Mabutu.A@mm	■	■					
Win32/Mytob.NJ@mm		■					
Worm:Win32/Antinny.AI	■		■	■	■		
Worm:Win32/Antinny.D	■		■	■	■		
Win32/Netsky.Z@mm	■	■	■		■		
Win32/Sober.Z@mm!CME-681	■	■	■	■	■		
Win32/Netsky.D@mm		■					
Worm:Win32/Antinny.AV	■		■	■	■		
Win32/Netsky.C@mm		■	■		■		
Win32/Netsky.B@mm	■	■	■				

Figure 4: Message characteristics of top 20 social engineering malicious software removed by the MSRT.

Malware Name	Machines Cleaned	Discovered	Generic Conversation	Free Stuff	Foreign Language	Physical Attraction	Mail Error	No Malware Found	Account Information	Malware Found on Computer	Accusatory	Current Events	Virus Alert / Patch Required
Win32/Netsky.P@mm	119,870	Mar-04	■	■		■	■	■	■	■	■		
Win32/Netsky.Z@mm	27,466	Apr-04	■										
Win32/Netsky.D@mm	23,575	Mar-04	■										
Win32/Netsky.C@mm	20,759	Feb-04	■	■		■	■	■					
Win32/Netsky.B@mm	18,604	Feb-04	■	■		■							
Win32/Netsky.W@mm	9,361	Apr-04	■					■					
Win32/Netsky.N@mm	6,149	Mar-04	■					■					
Win32/Netsky.Q@mm	5,696	Mar-04					■						
Win32/Netsky.AF@mm	5,623	Oct-04			■								
Win32/Netsky.X@mm	3,286	Apr-04	■		■								

Figure 5: Message content characteristics of top 10 Netsky variants removed by the MSRT.

severely limited. Similarly, the mood of the populace on any given day may affect their willingness to open a suspicious attachment.

The variables introduced by all these factors make it somewhat difficult to isolate the impact of a specific metric by choosing malware variants with relatively equal values for the remaining metrics. The best approximation we can likely make is to choose a set of variants from the same family and from within approximately the same time period. Malware variants from the same family usually share similar replication characteristics (# 4) but often differ in the specific replication characteristics (# 1) by sending different messages/attachments, etc.

For this analysis, we will use a set of Netsky variants that appeared in 2004. Netsky was chosen because, compared to other families, there is at least moderate agreement on variant naming between anti-malware vendors.

Figure 5 shows the message content characteristics of the 10 most prevalent Netsky variants, as removed by the MSRT.

With this comparison, it's interesting to see some patterns emerge, although there are clearly outliers. Specifically, Netsky.P is clearly the most prevalent variant of Netsky. Assuming all other factors are considered equal, we could point to the variety of social engineering techniques leveraged by Netsky.P as one of the possible reasons for the high prevalence.

With the remaining variants, there is a general pattern that more message content types translate to higher prevalence (with the exception of Netsky.Z and Netsky.D which buck this trend). Given that the prevalence figures are reasonably close for most other variants and that the trend is somewhat weak, likely the most interesting takeaway is relevant to Netsky.P.

CONCLUSION

Per the information provided in this paper, it is clear that malicious software that leverages social engineering techniques to infect a computer is a formidable threat to users today. On the positive side, as these threats have evolved, so have the protection and mitigation techniques used to guard against these threats. For example, significant changes have been made to *Microsoft Outlook* (attachment blocking, the object model guard) to decrease the number of email-borne threats. On a larger scale, the User Account Control feature of *Windows Vista* is intended to decrease the likelihood that a user executing malicious code can significantly and detrimentally affect the system.

That said, as long as operating systems and applications permit users to run code, users will have some, albeit obscure, methods for running malicious code. Thus, technology can

not be the sole solution to this problem. Instead, technology needs to be paired with education. The concept of leveraging user education to help reduce the impact of social engineering techniques is far from a novel suggestion. However, the industry may be able to offer more targeted forms of education and awareness in the future by conducting further studies on what attracts users to social engineering attacks on both a content level and a human level and then tailoring the educational aids accordingly. The discussion offered in this paper (and in [3]) is an early example of how we can initiate these studies.

REFERENCES

- [1] Braverman, M. (2006) Microsoft Windows Malicious Software Removal Tool: Progress Made, Trends Observed.
- [2] SearchSecurity.com Definitions: 'social engineering'. http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci531120,00.html.
- [3] Jordan, M., Goudey, H. The Signs, Signifiers and Semiotics of the Successful Semantic Attack. In Proceedings of the EICAR 2005 Conference. 2005, pp. 344 - 364.
- [4] McAfee Threat Center. http://www.mcafee.com/us/threat_center/default.asp.
- [5] Symantec Security Response Encyclopedia. http://www.symantec.com/enterprise/security_response/index.jsp.
- [6] Trend Micro Virus Information. <http://www.trendmicro.com/vinfo/>.
- [7] CA Virus Information Center. <http://www3.ca.com/securityadvisor/virusinfo/>.
- [8] Vgrep. Available from <http://www.virusbtn.com/resources/vgrep/index.xml?>.
- [9] Staniford, S., Paxson, V., Weaver, N. How to Own the Internet in Your Spare Time. In Proceedings of the 11th USENIX Security Symposium 2002.