Which Malware Lures Work Best? Measurements from a Large Instant Messaging Worm

Tyler Moore
Southern Methodist University
Dallas, TX, USA
tylerm@smu.edu

Richard Clayton
University of Cambridge
Cambridge, United Kingdom
richard.clayton@cl.cam.ac.uk

***Abstract*—Users are inveigled into visiting a malicious website in a phishing or malware-distribution scam through the use of a 'lure' – a superficially valid reason for their interest. We examine real world data from some 'worms' that spread over the social graph of Instant Messenger users. We find that over 14 million distinct users clicked on these lures over a two year period from Spring 2010. Furthermore, we present evidence that 95% of users who clicked on the lures became infected with malware. In one four week period spanning May–June 2010, near the worm's peak, we estimate that at least 1.67 million users were infected. We measure the extent to which small variations in lure URLs and the short pieces of text that accompany these URLs affects the likelihood of users clicking on the malicious URL. We show that the hostnames containing recognizable brand names were more effective than the terse random strings employed by URL shortening systems; and that brief Portuguese phrases were more effective in luring in Brazilians than more generic 'language independent' text.**

I. Introduction

Many online scams require the victim to visit a malicious website. In a phishing scam the website will impersonate a real website so that passwords and other credentials may be stolen. In a malware-distribution scam the victim will be enticed into visiting a website for a 'drive-by' infection or fooled into the conscious, but unwise, decision to click on a URL that downloads and executes a malware program.

In this paper we study a number of Instant Messenger 'worms' which spread by causing a putative victim to receive a message from one of their instant messaging 'buddies'. This message includes a URL which, if clicked, will download some malware onto the victim's computer. If the victim executes this malware then it promptly and automatically sends a message to the buddies of the newly infected person – thereby continuing the worm's spread.

As explained in Section II, the worms we studied all used an IRC (Internet-relay chat) channel for the command and control of the infection process, and we were able to monitor this channel for extended periods. From this monitoring we identified the exact message being sent to buddies (the 'lure') along with the rapidly changing location of the malware itself. Because some of the criminals chose to host their malware on web servers that had world-readable log files we could ascertain, over extended periods, how many potential victims were downloading the malware.

978-1-4799-8909-6/15/$31.00

Between May 2010 and July 2012, we observed nearly 63 million clicks from over 14 million distinct users. For a brief time, we had 'chanop' access on the IRC server along with access to the logs of a website set up by the criminals to host their malware. This allowed us to ascertain that 95% of the users who clicked on the lure executed the malware and became infected. Full details on the monitoring system and results are given in Section III.

Over the many months that we were gathering data the criminals used a variety of different lures and URL styles. In Section IV, we describe how lures with URLs resembling social networks experienced download rates two to four times as great as URLs relying on shortening services. Finally, since most victims from Fall 2010 onwards were from Brazil, in Section V we demonstrate that lures written in Portuguese attracted more clicks than those written in English and those written to be language-independent.

II. The Yimfoca Instant Messenger 'Worm'

A worm is a type of malware that spreads over a network from one computer to another by replicating itself. In this paper we consider a specific type of malware that spreads across Instant Messaging networks by automatically sending out messages that deceive humans into downloading and running the malware – after which their machines will commence the sending of further deceptive messages.

A. How the Malware Operates

In late April 2010, some malware started spreading over the Yahoo Messenger network and the interconnected Windows Live Messenger service. Users would see an instant message from one of their 'buddies' (their friends that they had enrolled into their address book) which said:

`foto ☺ http://example.com/image.php?user@...`

where the URL was for a copy of the malware and the email address (in the example abbreviated to just `user@`) was that of the recipient of the message.

The recipient's messaging software presented the URL as a clickable link and not surprisingly, since it came from a buddy, the recipients often clicked on the link. This caused a copy of the malware to be downloaded to their computer so that it would be executed.

If the recipient was running a version of the Microsoft Windows operating system (which a very high proportion of

them were) then just before execution commenced a standard warning pop-up dialog would appear to caution the user about the risks of running programs downloaded from the Internet. If the user was unwise enough to press the OK button then execution would proceed and the machine would be infected. Since the malware was invariably freshly minted it was never detected by anti-virus software, which would typically not be capable of recognizing it until several hours had elapsed. The malware was Windows-specific, so the small proportion of recipients on other platforms would have been unaffected.

The malware caused the recipient's browser to display a standard MySpace webpage which contained numerous headshots. The intention was clearly to make it look as if the buddy was inquiring if the victim's likeness was amongst these images. In the meantime the malware was making appropriate operating system changes to ensure that it would be re-executed on every boot and to disable anti-virus programs so that future updates to these programs would not cause the malware to be detected at a later time.

The malware then resolved a built-in hostname and used the resulting IP address to make contact with its Command and Control (C&C) system. The C&C for this malware used some privately operated IRC servers. The malware would log into the IRC server and use IRC commands to join a specific channel (named #jakarta).

On a regular basis (about every 800 seconds) all of the systems that were connected to the #jakarta channel would receive an IRC TOPIC command such as:

```
.m.e foto :D http://example.com/image.php?=
```

The first part of this command consists of instructions to the malware to send a message to all of the buddies of the victim and the second part is the text of the message to be sent (the :D is the encoding for a smiley face). The malware automatically replaces the = at the end of the URL with the email address of each relevant buddy, which it obtained from the contact details for that buddy as recorded within the address book of the Instant Messaging software.

Other instructions sent over the C&C channel would cause the malware to leave the `#jakarta` channel and join another channel. Once this happened no further messages would be sent to buddies, but commands on the new channel would cause additional software to be downloaded and run on the victim machine. At this time (April/May 2010) the additional software downloaded was usually a program that would force the user to answer a questionnaire whenever they visited a search engine or tried to visit a URL with pre-set strings within it – it is believed that the answering of this questionnaire caused an affiliate payment to be made, so this was a key part of monetizing the computer takeover.

B. Countering Yimfoca

The way that this malware operated with individualized messages coming from buddies turned out to be exceedingly effective and it spread rapidly. Initial reports appeared on Romanian web forums on 30 April 2010. A few days later a copy was sent to Symantec who named it 'Yimfoca' as a combination of Yahoo (the Instant Messaging system that was implicated in the distribution of the copy they received) and 'infocard.exe' the name of the executable it wrote to the victim's disk. This special name was a little misleading in that the malware is almost certainly just a Rimekud variant and indeed other firms identify it as PushBot-U.

The number of messages being sent by the Yimfoca malware was growing quickly as more computers became infected with estimates suggesting that over a million machines were infected at this time. Simple countermeasures were not especially effective because the malware had built-in resilience. If the site hosting the malware was taken down then it was merely necessary for the criminals to change the IRC TOPIC. If the IRC servers were taken down then the hostname that located the C&C could be caused to resolve to a new IP address for a new IRC server. If the hostname was suspended then the malware had several alternative hostnames built into its code, while new malware could be deployed at any time to update the hostnames.

In early May, Yahoo engineers deployed some automated blocking systems on their messaging system that detected rapidly trending URLs and discarded the messages. This almost entirely prevented the spread of the malware to Yahoo users but it continued to spread on the Windows Live Messenger system which was cross-connected to Yahoo.

In mid-June, a coordinated operation was carried out that ensured that all of the hostnames built into the malware were suspended and at the same time (within about an hour) all four of the then currently active IRC server machines were shut down. This coordinated response stopped the original Yimfoca worm from spreading further, and so it immediately died.

C. Later Instant Messenger Worms

Although the original Yimfoca worm was killed off, several further Instant Messenger worms were detected from July 2010 onwards. These later worms operated in very much the same manner as Yimfoca (with which they were clearly related) and they also infected many users. One adaptation that likely contributed to the later worms' success was that they spread over Facebook's messaging system in addition to more traditional Instant Messenger systems.

These later worms innovated further. They began using URL shorteners and moved away from the simple `foto ☺` lures. Once the worms changed to using a different shortener URL on every topic change (every 800 seconds), the automated systems at Yahoo that blocked repeated URLs became significantly less effective and so another set of coordinated takedowns was organized in June 2011. Although various further worms continued to be tracked into mid-2012 their impact was much reduced.

III. Worm Measurements

From late May 2010 onwards, a detailed set of measurements was made, with a view to understanding the impact of the worms and to assess the effectiveness of countermeasures.

A. Monitoring the C&C IRC Channel

The nature of the C&C IRC channel meant that it was entirely straightforward to create a Perl program to resolve an appropriate hostname, connect to the IRC server and join the `#jakarta` channel. The changing topic text could then be analyzed and the program could then automatically download copies of each new piece of malware for further investigation.

As we have noted above, the criminals who were deploying the malware operated several IRC servers in parallel – so it was necessary to monitor them all. Additionally, when the malware failed to resolve the first hostname compiled into it, some fall-back hostnames were tried – and thus the monitoring system had to resolve all known hostnames to identify all IRC servers. It was also essential to make regular checks of new malware samples to determine if new hostnames had been added.

IRC monitoring was in place from 27 May 2010 until the initial set of worms were disabled on 22 Jun 2010 and then after some patchy coverage, pretty much continuously from 26 Aug 2010 to 7 Jul 2012.

B. Monitoring Malware Downloads

The main information gleaned from the automated monitoring of the IRC C&C channel was the location of the latest version of the malware. The criminals registered brand new domain names at general purpose hosting providers. These domain names were used for short periods (usually just a few hours) before the hosting company suspended the site in response to abuse complaints – and the criminals moved on.

The criminals were not especially careful about how they set up their websites. In many instances from late May 2010 onwards, they left their Apache webserver logs world-readable. Accordingly, the Perl program that monitored the IRC C&C channels was extended to try and fetch the webserver logs for any websites that were being used to host malware.

The webserver logs give us full details (for the sites where the criminals left them accessible) of malware downloads. For the period from 27 May 2010 up until the take-down of the first set of worms on June 22 (a period of approximately 660 hours), we have logging data for 40.7% of this period.

However, not all of the downloads recorded in the logs result from people clicking on links – so we must exclude some automated fetching and monitoring events. For example, Facebook operates a screening system for URLs which means that the malware was invariably downloaded from a Facebook server IP address in parallel with the download by the user.

Therefore we excluded all downloads from Facebook IP addresses, all downloads from Yahoo IP addresses (for similar reasons) and from all the IP addresses we were able to associate with the monitoring systems of the anti-virus companies BitDefender and Trend Micro. We also excluded the downloads that were done by the criminals themselves when they checked that their website had not yet been suspended.

We identified all of these 'non-click' downloads by the simple expedient of looking for regularly recurring IP address ranges, and we are satisfied we have not overlooked any monitoring (by good guys or bad) that occurred at any scale.

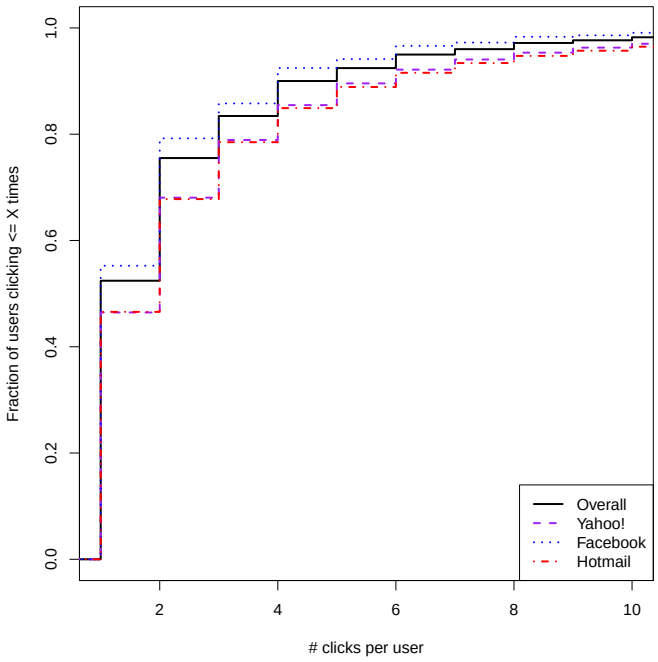

Fig. 1. Cumulative frequency distribution for the number of clicks (malware downloads) per identifier for all types of identifier (black), for Yahoo email addresses (purple), Hotmail/Outlook email addresses (red) and Facebook identifiers (blue).

Even after removing the non-click downloads, a problem remains in interpreting the raw data. It is clear that some people downloaded the malware on multiple occasions.

There are several reasons why the malware might be downloaded several times by one person. It may simply have been that several of their buddies became infected and sent them a message – and they clicked on all of these. Alternatively, some users might have been confused by the MySpace page that the malware displayed and clicked again to see if something different happened.

In order to understand how often multiple downloads occurred we exploit the fact that the messages received by any individual are unique to them. So far we have described this uniqueness as being the recipients' email address but for messages sent over the Facebook platform the parameter part of the URL is their numeric ID (a multi-digit number that can be used as part of a www.facebook.com URL to reach their Facebook pages). Since the full URL, including the parameter portion, is recorded in the web logs we were able to examine all the records with Facebook IDs and determine how often each ID was associated with a download of the malware.

The results of this are shown in Fig. 1 from which it can be seen that about half of the identifiers were associated with a single download request, 23% were downloaded twice, 8% were downloaded three times and 10% were downloaded five times or more.

It can also be seen that the Yahoo and Hotmail/Outlook distributions are similar but that there are fewer clicks per identifier for Facebook identifiers. We believe that the reason

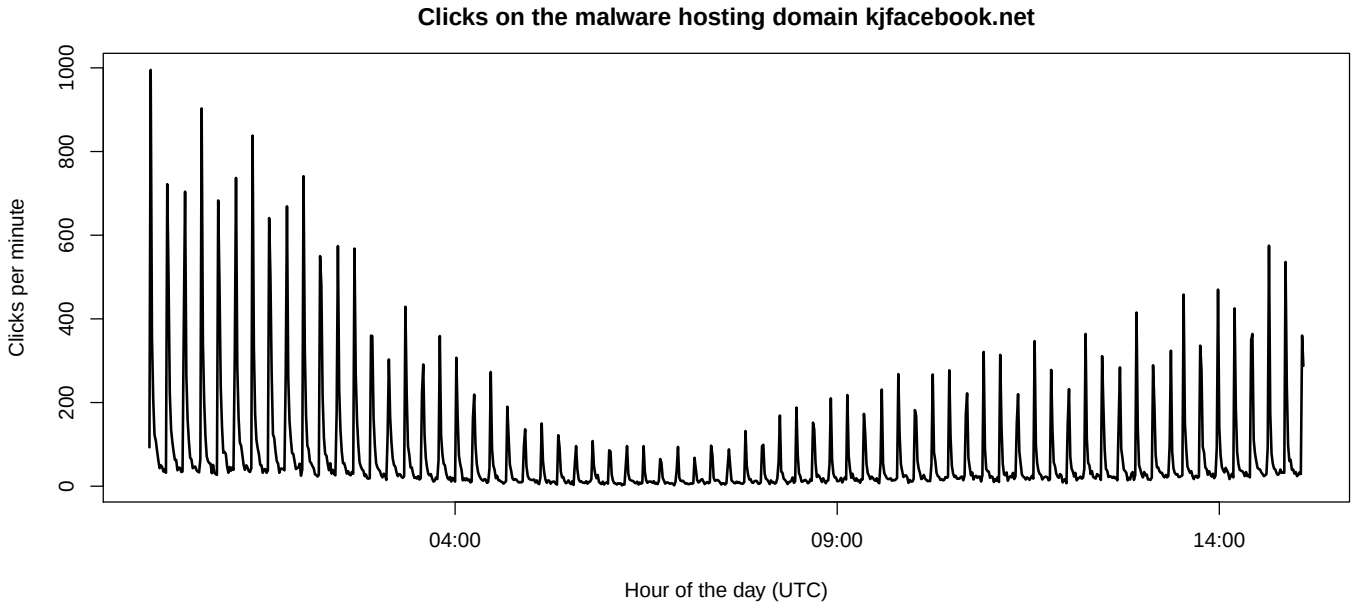

Fig. 2. Observed clicks per minute for the malware hosting domain `kjfacebook.net`.

for this is associated with the parallel download that Facebook performs for every URL – once the URL is detected to be malicious Facebook was in a position to protect their users by blocking further clicks.

The large number of multiple clicks means that processing the full set of download events may be misleading. Hence for all further analysis we only consider the first download associated with each identifier. However, not all downloads have a parameter recorded – for example some of the shorteners do include this in their redirection, although we can sometimes locate it in the HTTP Referer field. To address this, once a download has occurred from any given IP address we exclude all further downloads by that IP address for the following 48 hours. After all of this filtering our initial dataset of 62 800 890 clicks reduces to just 14 343 878 events.

C. Measuring 'Lure to Click' Time

We should note the rapidity with which people clicked on the links in the messages that they received. In Fig. 2 we plot the number of unique downloads per minute over a 15 hour period on 30 May 2010 from the then current malware hosting domain `kjfacebook.net`.

There were 10 118 downloads in the first hour, 8 232 in the second hour, and so on for the 15 hours before the domain was suspended. As can seen, the download events are extremely bursty, with the vast majority occurring within a few seconds of the IRC channel carrying a command to cause messages to be sent, with an exponential decay occurring thereafter.

Every 800 seconds the IRC command channel carries a further command to send messages to buddies and the number of clicks jumps up again. That is to say, most people see the message from their buddy and instantly click on the URL. A small number do not react immediately; perhaps they are away from their computer, or concentrating on a different task, but this graph shows us that it is reasonable to make the simplifying assumption that any given click is associated with the immediately preceding IRC channel event.

It can also be seen that the size of the peaks drops as time goes on before rising again. This is because the number of Instant Messenger users varies considerably at different times of the day, midnight UTC is generally the most active time.

D. Estimating the Infection Rate

Clickrates provide clear evidence about the 'social engineering' effectiveness of the lures (Foto, etc.) that were used. Since those lures varied over time it is possible to assess the extent to which some lures worked better than others and in later sections we will consider this issue at some length.

The figures for clicks do not tell us how many of the people who downloaded the malware were not running Windows and so could not become infected, nor does it tell us how many avoided infection by not pressing the OK button on the warning dialog – but some other data we acquired does provide evidence of the infection rate.

Besides joining the `#jakarta` channel the Perl program also joined all the other channels which the criminals used to control the malware. If one of these channels had not yet been created on any particular IRC server the act of joining meant that the Perl program created the channel and hence, because of the way that IRC works, the Perl program became the 'chanop' – which bestows special privileges and means that more information is provided about who is using the channel.

Usually the criminals used their 'ops' powers (controlling the whole IRC system) to remove the 'chanop' status from anyone outside the criminal gang. However, on a few occasions they failed to do this. The result was that for a handful of

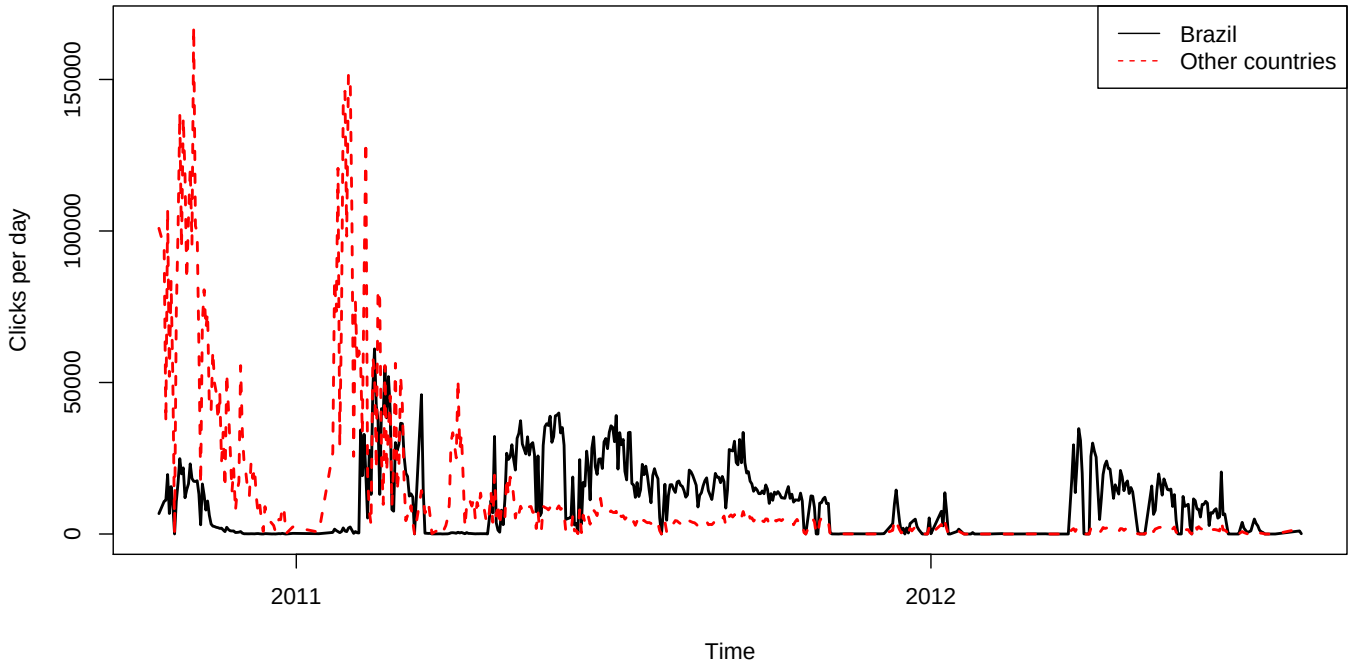

Fig. 3. Number of malware downloads per day for Brazil (black) and all other countries (red).

periods the Perl program was able to log all of the JOIN events that were reported to it as the chanop. That is, for these periods, we were able to log the identities of every malware infected machine that was commanded to join our channel.

This gave us a handful of snapshots of the infection rates. For example, between 2010-06-04 04:54:27 UTC and 15:15:44 UTC a total of 17 779 machines were commanded to join 'our' channel (1 717 machines/hour). During this period we had full coverage of the download logs and so we know from that data that there were 18 720 unique IPs that downloaded the malware. This is good evidence that during this period 95.0% of all of the downloaders became infected.

E. Estimating the Worm's Size

We start by considering the downloads we have recorded for the period May 27 to Jun 22 2010. After performing the de-duplication we discussed above we are left with 717 083 unique click events during the 40.7% of the total period for which we have data.

We can now scale this up in a naïve manner[1] and assume, as we have just shown to be true at one point in time, that 95% of clicks led to an infection. This leads us to conclude that at least 1.67 million users were infected between 27 May and 22 June 2010. This is an average infection rate over the whole period of 2 577 per hour. This is half as high again as our snapshot from the IRC channel, but this difference is unsurprising because the number of victims seen from this type of infection will not be evenly spread over time but will have exponential phases of growth.

The worm was active for almost the same duration before we started measuring on May 27, and for much of that time it spread unhindered over the Yahoo infrastructure, whereas by 27 May that spread was much inhibited. We therefore believe it is entirely plausible to estimate that the criminals who operated the first set of worms (Yimfoca and the others operating in that time period) caused rather more than three million machines to become infected.

F. Location of Victims

We now consider the download events for the period from September 2010 onwards (the second set of worms, which was interdicted in Summer 2011), for each of which we know the IP address. We use the Team Cyrmu Geo-Location service to determine the country associated with each IP address.[2]

We determined which countries the clicks were coming from and find that 43.3% of them came from Brazil. The details for the top 20 sources are in Table I.

In Fig. 3 we show how the number of downloads varied over time for Brazil and for all other countries. The various dips where there does not seem to be much activity occur when the criminals are hosting their malware at locations where we did not have access to the Apache webserver logs. It can be seen that the malware downloads by machines in Brazil initially lag those in the rest of the world, but from mid-2011 onwards almost all the activity is associated with Brazil.

In addition to the variance by day and by country, we also observed that the rate of downloading varied considerably by

[1]The sum is naïve because we make the simplifying assumption that our monitoring is randomly spread through the diurnal cycle, seen in Fig. 2. If we adjust for this then, depending which day we choose as a reference basis for the cycle, we get figures that are 20% to 80% higher.

[2]http://www.team-cymru.org/Services/ip-to-asn.html

Rank	Country	# Clicks
1	Brazil	6 124 878
2	Turkey	800 852
3	Thailand	444 560
4	Italy	323 656
5	Czech Republic	303 692
6	Taiwan	288 845
7	Colombia	282 502
8	Bulgaria	248 661
9	Saudi Arabia	245 843
10	Romania	236 008
11	Morocco	227 891
12	Mexico	209 673
13	France	187 693
14	United States	186 730
15	Germany	178 716
16	Peru	175 457
17	Spain	169 481
18	United Kingdom	165 650
19	Argentina	148 890
20	Portugal	129 010

TABLE I
TOTAL NUMBER OF CLICKS PER COUNTRY (TOP 20 SHOWN).

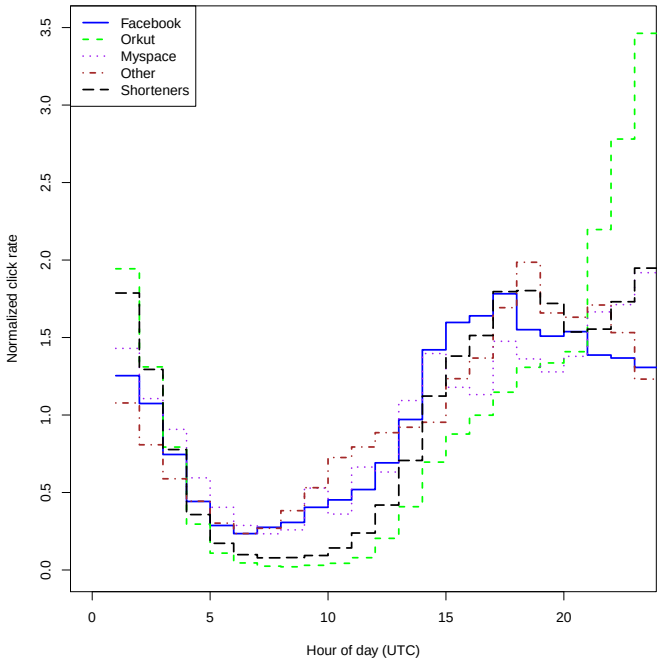

Fig. 4. Diurnal pattern of the proportion of clicks received throughout the day according to the type of domain included in the lure.

the hour of day. This might be expected because usage of Instant Messenger systems will differ between three in the morning and three in the afternoon. Fig. 4 plots the relative download rates as a function of the time of day in UTC. The plot shows how the download rates vary based upon the type of imposter domain used in the lure. We can see that regardless of which domain, if any, is being impersonated, there is a considerable drop between the hours of 0300–1000 UTC, with a peak around roughly 1700–0100.

Time	Lure	Rate
19:44:32	Foto ☺ http://fogz.eu/images886?=	83
19:57:57	Foto ☺ http://fogz.eu/images886?=	67
20:11:17	Foto ☺ http://fogz.eu/images886?=	72
20:24:36	Foto ☺ http://fogz.eu/images886?=	62
20:41:42	Foto ☺ http://fogz.eu/images886?=	91
20:58:10		60
21:10:47		63
21:24:03	Foto ☺ http://fogz.eu/images886?=	69
21:37:28	Foto ☺ http://fogz.eu/images886?=	67
21:51:04		72
22:04:22		79
22:08:13	Foto ☺ http://fogz.eu/images91?=	88
22:21:34	Foto ☺ http://justinloveis.net/album.php?=	132
22:34:54	Foto ☺ http://justinloveis.net/album.php?=	190
22:48:19	Foto ☺ http://justinloveis.net/album.php?=	115
23:01:41		106
23:15:09	Foto ☺ http://justinloveis.net/album.php?=	108

TABLE II
LURES USED ON FEBRUARY 14, 2011, WHERE THE TRAILING = IS REPLACED BY A MESSAGE RECIPIENTS' EMAIL ADDRESS.

IV. THE IMPACT OF URL SHORTENERS

As we remarked earlier, the criminals used a large number of different domains for hosting their malware. Because the domain name appeared in the messages that were sent, the criminals were generally of the opinion that it was important to choose plausible names. As will be recalled lures were initially something like `foto` ☺ and during this period domain names such as `msg-facebook.com`, `web-facebook.com`, `web-facebook.biz`, `newphoto-facebook.com` etc. were being used. Later on there was some occasional use of URL shorteners before, as we have also noted, they moved on to using nothing but shorteners.

Inspection of malware downloads during the transitional stage suggests that there was a negative impact on clickrates from using shorteners. Table II documents what happened over a short period on 14 February 2011 as the criminals switched from using a URL shortener (`fogz.eu`) to using the `justinloveis.net` domain name (which we assume was meant to trade upon a connection with the recording artist Justin Bieber – or perhaps 'inlove' relates to Valentine's Day). Since the `fogz` shortener pointed at exactly the same file on `justinloveis` there would have been no other difference in what a user saw, apart from the text of the link itself.

The Rate column shows the number of malware downloads per minute for the period after the given lure was sent out until the next lure sending time. It can be seen that the rate jumps markedly (by half as much again) once the lure changes from the generic URL shortener to a domain name. Note that some channel topics were blank so no messages were sent at that time, but users who did not examine their messages immediately would still have been in a position to click on the relevant link.

We can extend this analysis in a more principled way to the other domain names and shorteners that were used at various times. However, as Fig. 3 showed, the overall rate of download varied considerably over time, and even over the course of

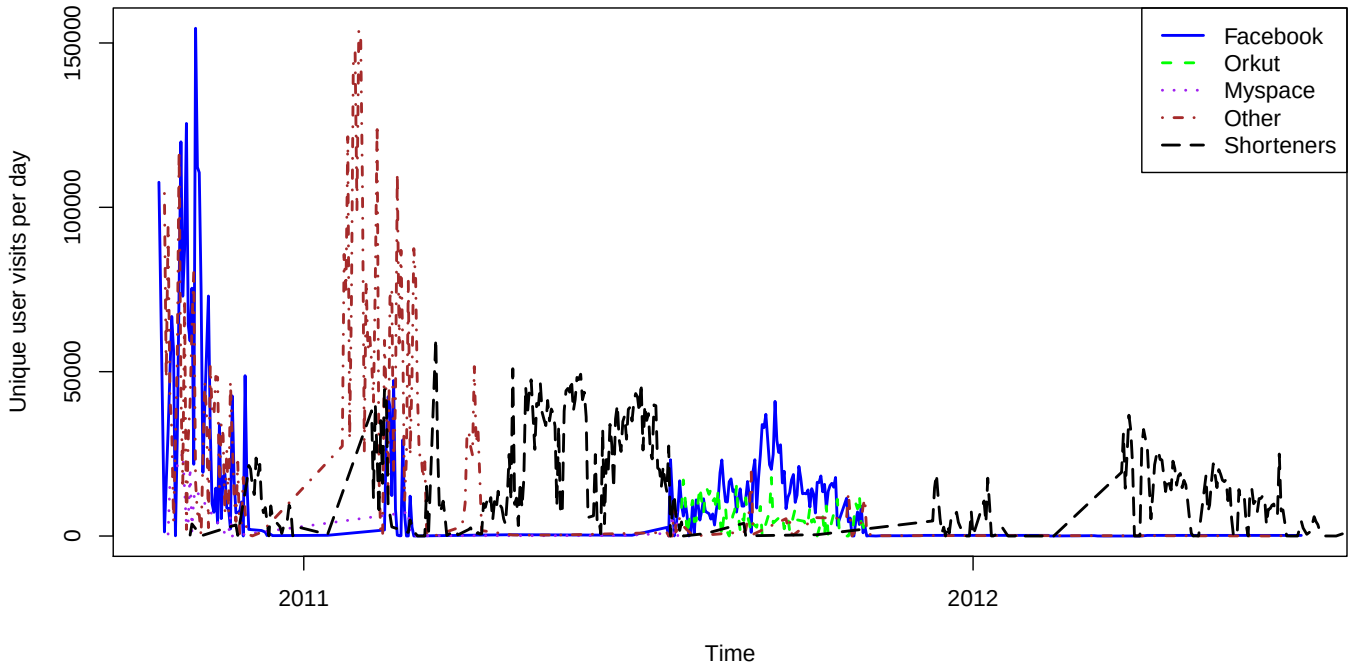

Fig. 5. Number of visitors downloading malware per day based on the type of domain included in the lure.

each day (Fig. 2 and Fig. 4). Therefore, we seek out periods of relative consistency in the worm's activity to enable more direct comparisons.

Fig. 5 once again plots the number of distinct visitors to download malware, but this time the results are separated by the type of imposter domain used by the perpetrators. It is evident from the graph that the criminals are experimenting with different strategies over time. In late 2010, the criminals used domains impersonating Facebook (e.g., `facebook-wifepic.net`) or in the 'other' category (e.g., `i-photoz.com` or `girlz-xxx.com`). In early 2011 they added URL shortening services, which they continued to use exclusively for months in mid-2011. Then, beginning around August 2011, they temporarily ditched the shortener strategy for Facebook and Orkut impersonations. Only a few months later, they gave up on that strategy, returning to URL shorteners exclusively.

We hypothesize that the criminals experimented in this way in response to pressure from defenders. Once Facebook began to crack down and suspend domain names, they tried URL shorteners. However, this approach was not as successful, so they switched back to impersonating services. Unfortunately for the criminals, this second attempt proved less fruitful than the first, so they eventually returned to using URL shorteners.

We now attempt to establish whether or not the use of impersonating domains was in fact more successful than shorteners. We focus on two time periods. First, we examine results between February and April 2011, when URL shortening services were used simultaneously with criminally-registered domains. Second, we examine the period August–October 2011 to compare the relative merits of domains impersonating Orkut and Facebook.

Because we are studying the impact of domains on luring victims, we wish to hold as much of the remaining variation constant as possible. Hence, we examine only those downloads made in response to the language independent lures of `foto` ☺ and `foto` ☺☺. These lures were also selected because they were used by impersonating domains of each type (e.g., Facebook, shorteners) at that time.

The results are presented in Table III. The table reports the number of impersonating domains used in each period, the total number of distinct visitors for each set of domains, and the download rate per minute for domains in each category (since some domains are used for longer periods than others).

However, as noted earlier, the domain names are each used at different times of the day, where download rates vary considerably. So we sum the total number of downloads per hour of the day over the period we are considering and calculate an adjustment value to bring the values back to the mean. This ensures that a download at 3am (which is 50–60% less likely to occur than on average) counts more than a download at 6pm (50–70% more likely to occur than on average). With these adjustment values we can determine the relative download rates for domain names referring to particular brands. These results are also presented in Table III.

For the first period, the normalized mean download rate of `foto` ☺ lures is 16 per minute, compared to 14 for other domains and 9 for shorteners. These findings are similar for the uncorrected figures. Both demonstrate that impersonated domains are moderately more successful than URL shorteners at luring in victims.

For the second period, the criminals had temporarily given

	Facebook	Myspace	Orkut	Other	Shorteners
			Period 2–4/2011		
# domains	13	1	–	65	17
# visitors (total)	140 149	11 625	–	920 355	424 835
# visitors/site (median)	11 324	11 625	–	10 978	3 842
Downloads/min. (mean)	22	45	–	16	10
Downloads/min. (med.)	6	45	–	11	3
Normalized rate (mean)	16	32	–	14	9
Normalized rate (med.)	5	32	–	11	3
			Period 8–10/2011		
# domains	51	0	37	0	0
# visitors (total)	152 949	–	136 265	–	–
# visitors/site (median)	2 991	–	3 142	–	–
Downloads/min. (mean)	7.1	–	6.8	–	–
Downloads/min. (med.)	3.4	–	3.0	–	–
Normalized rate (mean)	6.8	–	5.2	–	–
Normalized rate (med.)	4.7	–	3.0	–	–

TABLE III
RELATIVE EFFECTIVENESS OF BRANDING OF HOSTNAMES IN LURES. THE NORMALIZED RATE ROW GIVES THE NUMBER OF MALWARE DOWNLOADS PER MINUTE AS CALCULATED BY ADJUSTING THE ACTUAL RATE TO ALLOW FOR THE TIME OF DAY THAT THE HOSTNAME WAS IN USE.

up on shorteners, instead focusing on Facebook and Orkut impersonations. Here we note that the download rates are comparable, with a slight edge to Facebook.

V. THE IMPACT OF PORTUGUESE TEXT

From November 2011 onwards the criminals mainly used shorteners in the lures. However, they did vary the message that accompanied the URL and in many cases this message was in Portuguese. This variation allows us to compare the effectiveness of lures that are in people's native language with other lures that are essentially language independent (such as `foto`). Since the shorteners convey no semantic information we can be reasonably assured that any differences come from the rest of the lure.

To analyze the impact of using native language lures we consider the data for the period 1 January to 8 July 2012 – but only for the lures which contained shorteners. These lures were the most recent IRC channel topic (and hence the message that would be sent to buddies) for 10 394 520 seconds (120.3 days) during periods for which we have click data from website logs.

The lures were in English (e.g. `is this you?`) for 2.1% of the time, in Portuguese (e.g. `eu acho que é você na foto`) for 48.0% of the time and in a language independent style (e.g. `hahha foto`) for the remaining 49.9% of the time.

We can then count the number of clicks that each particular lure received and determine whether or not having the lure in Portuguese made any difference to the results. Fig. 6 plots the clicks in each hour for lures in Portuguese, English and language independent form.

As can be seen – lures in Portuguese receive far more clicks than language independent lures, without even allowing for them being slightly less prevalent. However, between 10am and 2pm UTC (7am and 11am Rio de Janeiro time) the language independent lures are clicked more often. This effect persists even when only Brazilian IP addresses are considered (they form 89% of the 1 227 315 clicks we are considering).

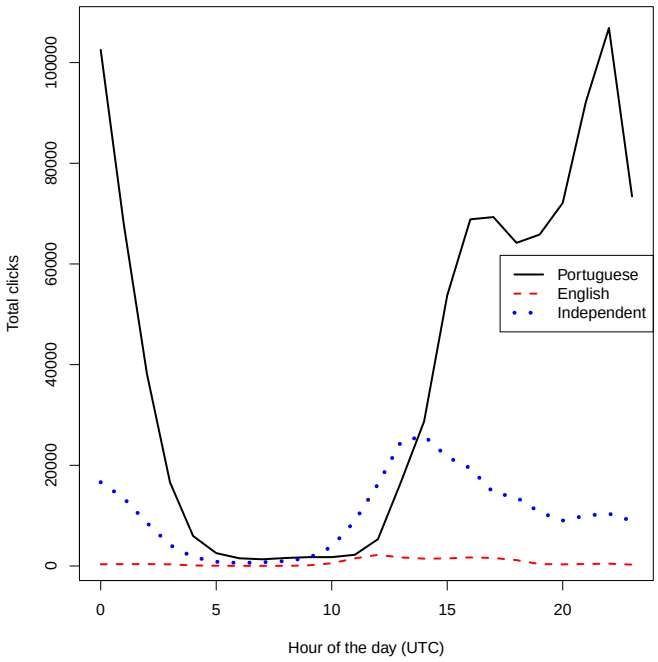

Fig. 6. Number of clicks on lures containing URL shorteners for the period 1 Jan to 8 Jul 2012. The lures are divided as to whether they are in Portuguese, English or a language independent form.

However the analysis we did above which showed that there was roughly the same level of exposure to Portuguese and non-Portuguese lures was for the whole of the period we are studying and we find that it differs by the hour of the day. The reason for this variation is unclear – it may be that the lures were set by more than one criminal and we are seeing that the Portuguese speaking criminals operated at different times.

Fig. 7 shows how the proportion of lures in each language varied over the day (once again we're only considering the periods for which we have click data). Superimposed on this is a line showing the percentage of clicks that were for Portuguese lures. Although this graph indicates that the more exposure the more clicks it also shows that this is insufficient to explain how language independent lures are more effective than Portuguese lures at some times of the day. The most likely explanation, in our view, is that different types of people use Instant Messenger at different times of the day. For example, we would expect a higher proportion of school-age children to be chatting with their buddies at times which are outside the working day.

So although there are slightly mixed results here – it is clear that at many times of the day, sending lures in Portuguese is a markedly superior strategy for the criminals to adopt.

VI. RELATED WORK

We are aware of very few studies which compare the effectiveness of lures. Moore et al. discuss how trending terms in search engine results are used by criminals to siphon off some of the traffic to their malware distribution websites [7]. McAfee regularly assesses which celebrities are the most

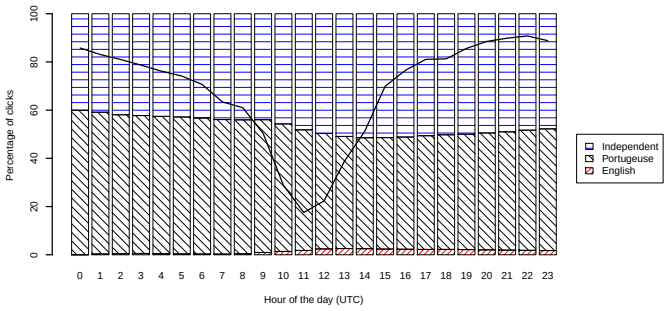

Fig. 7. Stacked bar graph showing how many lures are in English, Portuguese and language independent form at different hours of the day. The superimposed line shows the percentage of clicks that were associated with Portuguese lures.

dangerous to search for and publish a yearly report of their results – they measure whether the top search results for these celebrities are for pages which their SiteAdvisor system considers to be risky to visit [5].

Machine learning systems for detecting phishing emails inherently leverage the presence of lures – the features selected for the machine learning phase. For example, Bergholz et al. pick out the word stems "account, update, confirm, verify, secur, notif, log, click and inconvenien" and train their classifier accordingly [1].

In 2004 or so, when phishing first became a significant problem the criminals used to register domain names that resembled the brand that they were phishing, but they quickly switched to mentioning the brand elsewhere in the URL, if it is present at all. McGrath and Gupta provide a snapshot from late 2007 where 47% of PhishTank URLs have no brand name (albeit only 22% of phishing URLs were without a brand in the dataset they received from MarkMonitor – a 'brand protection' company that pays special attention to the brands they are paid to protect) [6].

Rather more work has been done on how to persuade people not to be fooled by lures. Kumaraguru et al. report on a large study of the effectiveness of training people to recognize phishing lures and phishing websites [3] and considerable work was done by the same research team to develop and refine the messaging of a 'landing page' that would replace a phishing page that a user was unwisely attempting to visit [4].

Gupta and Kumaraguru revisited the data collected from the landing page in 2014 [2] and considered how URLs had changed from 2008. They found that 2014 hostnames were twice as likely to contain more parts than in 2008 (that is that some criminals were including brands within the hostname string rather than in the page name).

The main limitation of the 2007 [6] and the 2008/2014 [2] data is measurement bias. For example, only a relatively small number of hosting companies replace phish with the landing page so in the 2014 data 35% of the phish were associated with a single hosting company, and hence with the small number of criminals who chose to use that hosting company. So although it's possible to make statements about what particular groups of criminals are doing at particular times, it is not practical to say from this whether the criminals' theories about what will be effective are reflected in actual numbers of victims.

VII. Conclusion

In this paper we have explained the mechanics of a very successful malware distribution scheme which spread over the 'social graph' of Instant Messenger 'buddies'. The initial 'lure' of `foto` ☺ combined with the presence of the recipient's email address in the malware URL proved extremely effective in causing people to download the malware. Once they had done that only a standard Windows warning dialog stood in the way to stave off infection. Our data shows that around 95% of those who downloaded the malware failed to heed the warning and became infected.

We have explained how we were able to monitor the Command and Control channels of the malware spreading mechanism. During the long periods where the criminals hosted their malware at websites with world-readable logs, we were also able to monitor the number of people who were 'socially engineered' into clicking on the message from their buddy and downloading the malware.

The criminals seem to have been experimenting with different branding for malware hosting domains and different lures to accompany the URLs they sent out. Our data from the website logs allows us to measure how much the effectiveness of their scheme changed as they did this.

We find that the criminals do slightly better using domain names that contain relevant brand names than when they use generic looking URL shorteners. When they consistently use shorteners, so that it is the rest of the lure that makes all the difference, we found that they were much more successful in getting Brazilians to click when the lure was in their native language of Portuguese.

It is important to understand what works in social engineering, not because we want the criminals to be more efficient. Rather, we hope it will inform the efforts made to train people as to what they ought to look out for.

Doubtless, there is a role here for laboratory experiments with carefully controlled conditions, with variables carefully changed just one at a time to identify the factors that make a difference. But there is also benefit in observing what happens in the real world as cybercrime unfolds. Indeed, even amongst the chaos of over sixty million download events, we have successfully uncovered 'natural experiments' in criminal activity that help illuminate several key behavioral factors and quantify their impact.

Acknowledgments

The authors are funded by the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHSS&T/CSD) Broad Agency Announcement 11.02, the Government of Australia and SPAWAR Systems Center Pacific via contract number N66001-13-C-0131. This paper represents the position of the authors and not that of any of the aforementioned agencies.

REFERENCES

[1] A. Bergholz, J De Beer, S. Glahn, M. Moens, G. Paaß, and S. Strobel. New filtering approaches for phishing email. J. Comput. Secur. 18(1), pp. 7–35, 2010.

[2] S. Gupta and P. Kumaraguru. Emerging Phishing Trends and Effectiveness of the Anti-Phishing Landing Page. In: Proceedings of the Ninth APWG eCrime Researcher's Summit, Birmingham, AL, 2014.

[3] P. Kumaraguru, J. Cranshaw, A. Acquisti, L. Cranor, J. Hong, M.A. Blair, and T. Pham. School of phish: a real-world evaluation of anti-phishing training. In: Proc. 5th Symposium on Usable Privacy and Security (SOUPS '09), ACM, New York, NY, USA, 2009.

[4] P. Kumaraguru, L. Cranor, and L. Mather. Anti-Phishing Landing Page: Turning a 404 Into a Teachable Moment for End Users. Conference on Email and Anti-Spam (CEAS), 2009.

[5] McAfee Inc. McAfee Reveals Jimmy Kimmel As the Most Dangerous Cyber Celebrity of 2014. Press Release, 1 Oct 2014. http://www.mcafee.com/us/about/news/2014/q4/20141001-01.aspx

[6] D.K. McGrath and M. Gupta. Behind Phishing: An Examination of Phisher Modi Operandi. Workshop on Large-Scale Exploits and Emergent Threats (LEET'08), Usenix, 2008.

[7] T. Moore, N. Leontiadis, and N. Christin. Fashion crimes: trending-term exploitation on the web. In: Proc. 18th ACM Conference on Computer and Communications Security (CCS '11), ACM, pp. 455–466, 2011.