

MENACE 2 THE WIRES: ADVANCES IN THE BUSINESS MODELS OF CYBER CRIMINALS

Guillaume Lovet
Fortinet, France

Email glovet@fortinet.com

ABSTRACT

Today, the profits generated by cybercrime worldwide are somewhere between \$50 billion and \$100 billion per annum, flirting with the revenues yielded by the ‘historic’ business of trading illegal drugs.

However, as the public becomes aware of the situation, user education and global security policies tend to improve as well. To sustain profitable balances – or simply to optimize their gains – money-driven cyber criminals are pushed to innovate, to polish their social engineering methods and to go as far as taking physical action to implement their business logic. While companies are not spared, their targets of choice remain the average user. You, me, anyone.

While ‘Dirty Money on the Wires’ [1] was a snapshot of the most ‘traditional’ business models among the cyber-underground scene over the past two years, this paper will go deeper underground, closer to the culprits: based on quantified data, light will be shed on new – or anticipated – business models, following the evolution of cyber criminals as we are entering the Web 2.0 era, and as borders between crime and cybercrime become thinner every day.

INTRODUCTION

This paper is the sequel to ‘Dirty Money on the Wires: the Business Models of Cyber Criminals’ (a.k.a. ‘DMOTW’) [1], which was presented at the 2006 Virus Bulletin Conference (VB2006). Although previous reading of DMOTW is recommended to anyone wanting a clearer view of the cybercrime scene (profiles, marketplace, currency, channels etc.), it is not a formal prerequisite for this paper. The two papers do not overlap, save for the following, essential definition:

Cybercrime is a term used broadly to describe criminal activity in which computers or networks are involved, regardless of the level of such an involvement (source, target, means etc.).

While DMOTW focused more specifically on cyber criminals involved in schemes that rely tremendously on the internet, this paper explores some of the schemes that walk the (blurry) line between cybercrime and traditional crime; but above all, it tries to connect the dots in today’s monetized threat landscape, which closely follows the evolution of our online habits towards the so-called Web 2.0 era. As often as possible, the business models presented will be quantified with real data, and ‘in-the-wild’ examples will be given.

1. MASS INJECTIONS

Mass injections are a relatively new trend, building on the foundations of mass defacements, which have kept several

generations of (very) young hackers busy for countless hours. As will be demonstrated in this chapter through the MPack case study, mass-injection-based business models can be very lucrative.

1.1 A bit of history

Defacing a website simply involves hacking into the host server and replacing its index page with a (generally heavily customized) ‘you were hacked’ page; in addition to the ubiquitous ‘defacer group’ emblem, the defacement page usually sports a more or less subtle message, ranging from a plain ‘f*ck the w0rld’ written in ‘leet speech’ to more constructed political statements, and going through the very typical web admin taunting line.

In general, defacing is not destructive for the data sitting on the web server (the original index page is not deleted, and is sometimes even linked from the defacement page), although it can have a negative impact on the image of the company whose website was hacked, and of course on its productivity in the case of a commercial site.

Figure 1 shows a typical defacement page, hosted for the sake of, well, ‘history’ on a defacement mirror.

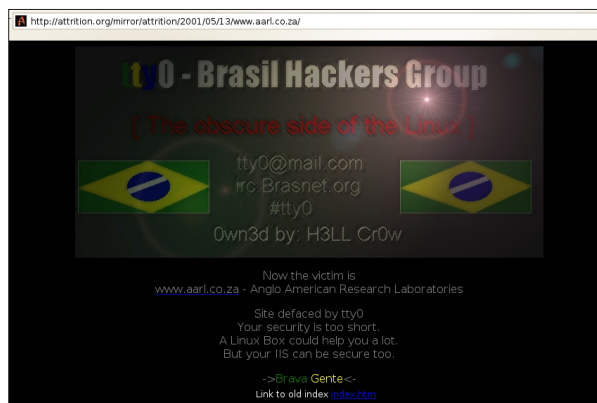


Figure 1: Dark gfx, a national pride touch, a good deal of leet speech, admin taunting, and Linux preaching: the defacement page paradigm.

It is worth mentioning that defacement mirrors have been controversial since the early 2000s, when they were often considered as a tremendous catalyst to the competition among defacer groups (some mirrors even held ranking rosters of the most proficient groups). In such a context, what is even more ‘rewarding’ and exciting than a defacement is a mass defacement, that is to say compromising a server that hosts several websites and replacing all the index pages with a tiny script.

Defacements often serve purely as a means of expression of national or ethnic pride for youngsters of minorities, and in essence, are not motivated by financial gain. However, lately, an interesting evolution has been witnessed involving cases where the defacers have added a means to generate a few dollars in addition to asserting their ethnic or religious identities. This can be observed in the ‘live’ defacement shown in Figure 2.

Of course, the ad box at the bottom leads visitors to an online lottery site, which rewards the defacers’ affiliates (in this case the hacker group) on a pay-per-click basis. Given the generally low frequenting of the defaced websites (microsoft.com



Figure 2: Please click to help fund the cyber-guerilla (our soldiers need new supplies of coffee and cigarettes).

certainly does not have the same level of security as bikersunited.It, thus is less likely to be defaced) and the short window of opportunity before the index page is put back online again, this practice is certainly not draining insane amounts of money on the wires. However, some more business-minded cyber criminals pushed this logic an inch further, effectively implementing the switch from mass-defacement to mass-injection, as was most famously witnessed in the MPack case.

1.2 The MPack case

Should you gain write permission on the index page of a large number of websites at once (on account of virtual hosting), there is indeed a wide range of more lucrative actions to attempt than plain defacement. This is probably what the gang behind the MPack attack found out and decided to implement.

The facts are rather straightforward: back in June 2007, more than 8,000 Italian sites were compromised, in what seemed to be a main hosting server hack. At the time of writing, the attack is still under investigation, but given that 90% of hacked sites were hosted by the same company, the one-attack-to-hack-em-all scenario is credible. Either way, the fact is that all those sites were poisoned with a malicious and invisible IFrame, silently redirecting visitors' browsers to an MPack server. This can be observed in the HTML source code of a compromised page shown in Figure 3.

The source displays an IFrame HTML tag, defining a window of 1x1 pixels (thus invisible to the user) in the compromised page, the effect of which is to make visitors' browsers request the URL blurred above; this URL leads to the malicious MPack server, via several hops.

MPack, sometimes also referred to as 'Webattacker II' is a piece of software edited and sold on more or less underground forums by a gang of Russian 'coders' for about \$700. It is essentially a collection of PHP scripts performing user agent recognition and serving the ad hoc exploits to visitors' browsers. Should a browser that just got served not be up to date, the exploit succeeds and a customizable trojan is downloaded and installed silently on the victim's computer.

```
?><!-- ~ --><iframe src='http://bl0cker.info/st/go.php?sid=1' width=1 height=1 style='visibility: hidden;'></iframe>
<iframe src='http://bl0cker.info/st/go.php?sid=3' width=1 height=1 style='visibility: hidden;'></iframe><!-- ~ -->
```

Figure 3: HTML source of the injected IFrame.

Attacked hosts: (total/uniq)	
IE XP ALL	87093 - 79152
QuickTime	37 - 34
Win2000	3953 - 3393
Firefox	18028 - 17796
Opera7	25 - 25

Traffic: (total/uniq)	
Total traff:	112525 - 102044
Exploited:	13765 - 10705
Loads count:	14103 - 5224
Loader's response:	102.46% - 48.8%
User blocking:	ON
Country blocking:	OFF

Country	Traff	Loads	Efficiency
IT - Italy	76625	11539	15.06
ES - Spain	8042	466	5.79
US - United states	3877	133	3.43
DE - Germany	2927	147	5.02
FR - France	1967	73	3.71
GB - United kingdom	1670	60	3.59

Figure 4: Notice the strong password.

Given the number of poisoned sites, and the fact they covered a wide range of categories, from hotels to designer clothes, the number of innocent visitors who were infected while browsing was significant, as can be seen in Figure 4, which shows the administration interface of the main MPack server involved in the attack.

This screenshot was taken a few hours before the server was made unreachable. According to the statistics it displays, during the few days the attack lasted, more than 100,000 individual hosts were redirected from the poisoned pages to the malicious server, and over 10,000 of them were infected. Keeping track here of the malware installed on exploited users' machines has little interest: either way, once a trojan is installed, it suffices to assume that the machine is under total control of the cyber criminals, for additional components can be installed at any time via various command and control channels.

The most important point here is that the attackers crafted a botnet of 10K hosts in a blink, and resorted to a strategy that reduced the impact factor of human resistance on the attack tremendously, since infection needed no user interaction; moreover, the poisoned sites were perfectly 'legit'. This is all the more interesting because of the fact that traditionally, humans are considered the weakest link in the security chain, and are often the target of (sometimes very lousy) social engineering attempts in order to 'break in'. To that regard, this case (albeit not the first one) can be considered as an advance in the logic employed by cyber criminals.

The business logic behind the whole attack is still formally unknown at the time of writing. Several scenarios can be considered, but in any case, the infected hosts can be used to

relay spam. The business model could then be congruent to the breakdown below:

Costs

- MPack software: \$700
- Compromising a host company server hosting thousands of sites: some hacking skills or \$10,000 (assuming 0-day)
- Script inserting IFrames into each page: little skill, or about \$50

The tricky part here is of course the host company hack. There is no universal rule, but if we assume that high-profile hosting companies' servers are fully patched, it is still possible to compromise one with a so called 0-day exploit (i.e. a non-public exploit for which no patch exists). On the cybercrime black market, the price for such exploits varies from \$5,000 to \$50,000.

Profits

- Assuming:
 - 10,000 infected computers used as a spam relay
 - Each one sends 100K emails before being blacklisted on RTBLs
 - Advertisers pay 0.03 cents per email
- 10,000 x 100K x 0.0003 = \$300,000 (one shot)

Summary

- Total costs: \$10,750
- Total profits: \$300,000
- Gain: \$289,250
- Productivity index (Profits/Costs): 27

As a matter of course, it is possible to boost the productivity of the model by multitasking the infected machines, and use them not only for relaying spam, but also for adware planting, banking trojan planting, click fraud, etc.

2. THREATS 2.0

As we are entering the Web 2.0 era, the threat landscape is evolving accordingly. Providing detailed information about the 'Web 2.0' concept itself is not within the scope of this paper, however, it is essential to point out the main idea behind it, as seen in a *Wired News* article [2]:

'...seemingly every aspect of our data [is] moving toward online apps and away from the traditional desktop model.'

As a simple example, this paper was fully written and edited online with *Google docs*. Security-wise, the main consequence of this shift of data and shift of working habits from the desktop to online applications is the expected rise in online identity theft attacks. Impersonating the user of an online application may give the attacker access to this user's data (related at least to this precise application), and may grant him the opportunity to perform actions on behalf of the impersonated user.

Depending on the online app security model, and how authentication and user sessions are implemented, different types of attack would be effective. While plain user-side trojanning remains the most robust strategy, currently the two weapons of choice for 'attackers 2.0' seem to be XSS and

CSRF. Again, in-depth descriptions of the technical details of these attacks is out of the scope of this paper, although the following short definitions may be useful:

- Cross site scripting (XSS) is a type of computer security vulnerability typically found in web applications which allows code injection by malicious web users into the web page viewed by others. Examples of such code include HTML code and client-side scripts. An exploited cross-site scripting vulnerability can be used by attackers to bypass access controls such as the same origin policy. [3].
- Cross site request forgery (CSRF) works by exploiting the trust that a site has for the user. Site tasks are usually linked to specific URLs (for example: `http://site/stocks?buy=100&stock=ebay`), allowing specific actions to be performed when requested. If a user is logged into the site and an attacker tricks their browser into making a request to one of these task URLs, then the task is performed and logged as the logged in user. [4]

In a nutshell, XSS exploits the trust that a client has for the website, while CSRF exploits the trust that a website has for the user. Typically, when a website suffers from an XSS vulnerability, an attacker can get his/her victim's browser to post the victim's session cookies to an attacker-controlled channel, and use these cookies to hijack the victim's session on the targeted site.

When a site is vulnerable to CSRF, if the victim visits a specifically crafted attacker-controlled site, actions defined by the attacker will be executed on behalf of the victim on the vulnerable site (provided the victim is logged in).

Obviously, if we add plain old user-account phishing to these two base weapons, and a tad of automation, attackers have a solid arsenal at their disposal to start milking the Web 2.0 cow. Various implementations of such attacks spotted in the wild are examined next; then existing or possible associated business models are formally devised.

2.1 Social worms

Social worms, also known as phisher worms, are parasites of social networking sites, and as such, are conceptually and practically linked tightly to those. In November 2006, the *Fortinet* Global Security Research Team discovered such a social worm that was scouring the ultra-popular community and networking site, *MySpace.com*. Basically, the worm would pop up in *MySpace* users' mailboxes in the form of a 'bulletin' message; as can be seen in Figure 5. The message

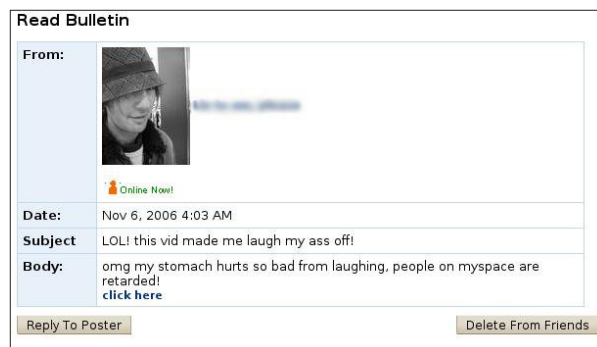


Figure 5: Social engineering 101. Simple but effective.

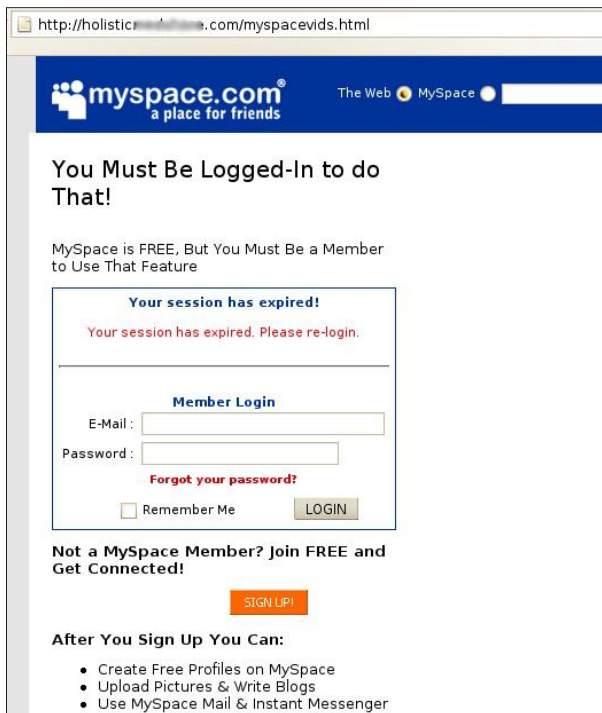


Figure 6: Rogue login page. Notice the URL in the address box.

text would entice the user to click on a link pointing seemingly to an amusing video.

Of course, the ‘click here’ link directed users to a phishing site mimicking the *MySpace* login page, rather than to the advertised video, as can be seen in Figure 6. Anyone entering their credentials into that rogue login page, hoping to see the video (which happens frequently on *MySpace*), would have had their account details stolen. But that was not all: a server-side program on the rogue server would also then distribute the initial message (carrying the rogue link) to all the contacts of the freshly phished user, hence effectively propagating the phisher worm throughout the community.

So, what we have is a creeping phish (a phish that spreads automatically, using worm-like features). Given that the average *MySpace* user has between 100 and 500 friends, this social worm probably harvested thousands to millions of *MySpace* accounts.



Figure 7: Infected profile – the whole page points to the same link.

2.2 Social worms++

A few months later, in March 2007, a new instance of the phisher worm was spotted in the wild, resorting to a tremendously enhanced propagation strategy and playing a pretty cunning mind trick to improve its phishing ratio. Infected profiles looked normal at first sight, but cautiously observing the address in the browser’s status bar revealed an anomaly: while this address normally changes as the mouse pointer navigates over different links, in this instance it remained the same, wherever in the page the mouse cursor was positioned. This can be observed in Figure 7 (the status bar was highlighted in red).

A look at the link the page is pointing to reveals that it resides on *Myspace.com*. However, the link is a redirector. Upon clicking on it, Myspace.com redirects users to the URL passed as the ‘redirect’ argument (extreme right in the status bar in Figure 7). Of course, the link leads to yet another fake login page (Figure 8).

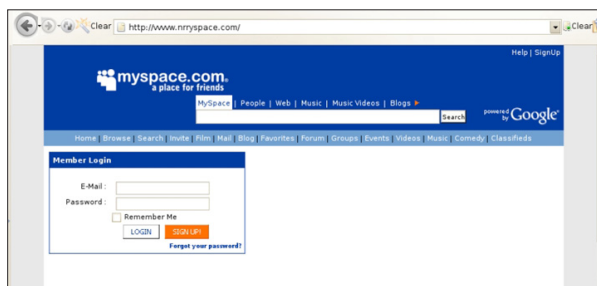


Figure 8: Rogue server phishing page.

The domain name is particularly interesting, as it attempts to trick the targeted user’s mind into thinking he’s on a genuine Myspace.com page: *vvvww.nrryspace.com*, or, when inserting spaces between characters: *v v v w . n r r y s p a c e . c o m*. Dirty.

Granted, the culprits now detain the credentials of the targeted users, one may still wonder how the hackers behind this scheme made the whole page clickable. The answer can be found in the HTML source of the infected page, which contains the following tags in the ‘about me’ section:

```
<a href="http://deict.myspace.com/event.ng/
[...]Redirect=http://vvvww.nrryspace.com/">

</a>
```

Essentially, these tags display an image whose size is 950 x 1,000 pixels (hence covering the whole page!), and whose source is a transparent gif sitting at *http://x.myspace.com/images/clear.gif*. The image is, you guessed it, clickable, and sends users to *nrryspace.com* via a *MySpace.com* open redirector.

In a nutshell: hackers (or rather, the program sitting on the rogue server) covered the page of the infected users with a clickable transparent image, in order to attempt to infect more users (who will in turn infect more users – this is a worm). Injecting this malicious code is made possible because *MySpace* allows users to embed certain HTML tags (essentially `<a>`, `` and `<div>`) in various parts of their pages. This is a Web 2.0-ish feature, and partly why *MySpace* became so popular.

Since it resorts to a blend of malicious strategies, including tricky user-provided HTML, phishing, automation, redirectors and mind tricks, this threat may effectively be called a best-of-breed piece of malicious set-up. It is worth noting that the notion of ‘malware’ is out-driven there, since one bit of the malicious code sits in the form of malevolent HTML on the infected user’s page, while the other bits (the phishing page and the engine in charge of posting the malicious HTML to the phished user’s page) sit on the rogue server, and that part of the threat lies in the domain name registration phase.

This phisher worm potentially has a greater impact than the one previously described, for the phishing page is accessible not only by ‘friends’ of the infected users, but by anyone visiting a public profile.

2.3 XSS/CSRF worms

Back in 2005, a *MySpace* user called Samy exploited an XSS flaw in the site, and combining it with CSRF, found himself with over one million ‘friends’ within 20 hours. Technical details of the attack can be found in [5], but in a nutshell, he managed to bypass *MySpace* keyword filtering so as to embed JavaScript in the editable parts of his profile page. This JavaScript would send an ‘add friend’ request to him from any user viewing his profile, then would add itself to this user’s own page – hence propagating exponentially like a worm. This was dubbed ‘the Samy worm’.

More recently, in December 2006, the so called ‘Quickspace worm’ was unleashed, again on *MySpace* (this is the downside of being the number one social networking site). Exploiting the facts that 1. *MySpace* would allow embedding of *Quicktime* movie files in users’ pages and 2. *Quicktime* movie files allow the embedding of JavaScript in a so-called ‘track’ of the file, the worm, once again, propagated malicious JavaScript from profile to profile (viewing an infected profile was enough to get infected). While Samy’s worm was purely a geeky joke, meant to send Add Friend requests to him on behalf of each infected user, Quickspace’s intents were clearly malicious: the evil JavaScript would cover the *MySpace* header section on infected profiles with a fake one leading to a rogue login page. The goal was therefore to leverage this XSS vulnerability to phish as many *MySpace* accounts as possible.

The lesson to be learnt from these two practical cases is threefold:

- XSS and CSRF vulnerabilities were discovered and exploited in Web 2.0 enabled sites, and given how hard such flaws are to spot, this will likely happen again in the future.
- The good news is that XSS/CSRF alone cannot usually be used to steal user accounts automatically – for instance on *MySpace*, should a user request a registered email change, it must be confirmed with an activation code sent to the former email (and of course a password change prompts you to enter the old password). Proof is that the Quickspace worm, although basically having full JavaScript execution privilege in infected users’ browsers, had to resort to phishing to actually steal their credentials.
- The bad news is that XSS/CSRF alone can impersonate most other actions infected users may perform on the site. And this is theoretically enough to achieve what cyber criminals are willing to achieve here, as will be detailed in next section.

2.4 The business logic

Indeed, the case studies above leave one question: for a business-minded cyber criminal, what is the point in gathering thousands of *MySpace* account credentials?

Actually, spam emails have become so common in our mailboxes that their click-through rate fell to unimpressive values, sometimes as low as 1 click out of 100,000 emails sent; spammers therefore tend to look for new spam supports.

Enter *MySpace*, with more than 106 million accounts (as of September 2006), each account bearing a ‘comments’ section. Comments are messages left by ‘friends’ (i.e. people who either requested or approved friendship with you). Each comment is displayed directly on the recipient’s main page and can be seen by all visitors browsing the profile (unless comment approval is requested).

MySpace comments are therefore an appealing new medium for spammers. However, spamming *MySpace* accounts are way more difficult than spamming mailboxes:

- One must be someone’s friend to send him/her a message, involving manual steps to build a friend network;
- Each comment can be tracked back in the case of abuse, resulting in banning.

Therefore, the most straightforward way to spray spam all over *MySpace* is to steal existing accounts (or hijack active user sessions) and post on behalf of the impersonated users.

Figure 9 shows an ad posted by a ‘friend’ of this account, posing as a legitimate comment and enticing the reader to ‘click here’ – which, of course, gets redirected, in this case to an adult site (third comment).

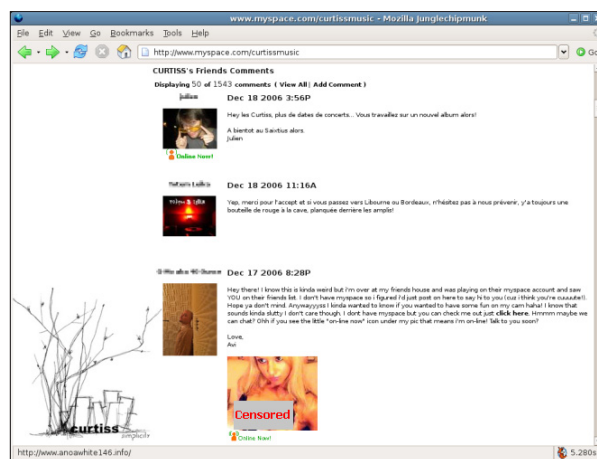


Figure 9: The third comment is a spam.

A closer look at the spam-like comment in Figure 10 reveals that it makes heavy use of social engineering.

1. Note how the message mimics the real *MySpace* layout: a catchy picture plus the ‘online now’ indicator right below (meaning there is someone behind the screen). This indicator is a copy of *MySpace*’s one (which normally sits below the sender’s image, on the left of the comment).
2. Please delight yourself with the cunning comment. Social engineering artists have long understood that lust



Figure 10: Social engineering, advanced course.

and vanity are very exploitable human flaws. This is a perfect demonstration.

Now, whenever someone clicks on that link, the spammer gets rewarded by the adult site. Depending on the affiliate program, the rates per click vary significantly, but if we consider that \$0.01 per click is the minimum possible rate on most programs, and that certain *Google AdWords* cost up to \$80 per click to advertisers [6], it is reasonable to assume a rate of \$0.05 per click for your average porn site – although some adult-related affiliate programs generally advertise higher rates. As a side note, business-wise, it makes sense for a site to spend \$0.10 per click if its conversion ratio (i.e. the percentage of visitors actually buying something) is 1% and its average profit per-buy is \$30.

As an example, let us consider a spammer who, thanks to a social worm, silently ‘owns’ a mere 6,000 accounts. It is generally accepted that on average, users have about 75 friends on social networking sites (that is to say, an owned account can post comments similar to the spam depicted in Figure 10 to 75 accounts) [7]. However, since friend lists may overlap let us assume that this pool of 6,000 accounts allows the spammer to reach 60,000 individual accounts. *MySpace* having close to 1.5 billion page views per day [8] and probably about 50 millions active users [9], the average number of page views per account per day is 30.

Thus:

- The 60,000 ads posted will be viewed 1,800,000 times, daily.
- Assuming a click-through rate of 5% (that is to say out of 100 people viewing the page, five will click on the spam comment – which is probably an underestimate, given the particularly refined social engineering speech and the profile of *MySpace* surfers), this leads us to 90,000 daily clicks.
- 90,000 daily clicks means a raw profit of \$4,500, daily, assuming a rather low reward of \$0.05 per click.
- This corresponds to \$135,000, monthly.

As a matter of course, this quantification is debatable: how many individual accounts can be spammed from a given number of stolen accounts is unclear, due to the tendency for friend lists to overlap. So whether the click-through rate is sustainable over time depends largely on how fast each page’s individual viewers are renewed (i.e. among the 30 daily page views, how many of them haven’t already seen the ad during a previous visit); moreover the number of page views includes viewing other sections of the site as well as the front page (e.g. the pictures section), etc. Still, this gives a good idea of the amplitude and profitability such business models can

reach, especially with a higher number of stolen accounts (6,000 being relatively small, after all).

As for XSS/CSRF worms, we can safely rely on Samy’s worm propagation figures: within 20 hours, it infected more than 1 million individual profiles [10]. Theoretically, this means the worm could have posted (on behalf of infected users) ads to at least 1 million different profiles. Sticking to a 5% click-through rate, and a \$0.05 per click rate, this would have generated at least \$75,000 during the first 24 hours. And probably over half a million dollars in less than a week.

However, given the large amounts of money involved, and the financial traceability via the affiliate programs, the risks are high; based on the cybercrime scene structure described in DMOTW, it is likely that social networking site phishers do not use their stolen accounts pool for their own profit, but rather rent them to hardcore spammers, as botnet herders would do. Hence the following business model:

Costs

- Assuming:
 - Target: posting an ad every week (so that it is always on the front page) for a month to 60,000 individual profiles
 - Price to pay for each posted ad: equals 10 times the average price to pay a bot herder for sending out one spam email (~ \$0.003)
- Renting the services of a social networking site phisher: 60,000 x \$0.003 x 4 = \$720 per month

Profits

- Assuming:
 - Each ad is viewed on average 30 times per day (equals the average daily page views per profile on *MySpace*)
 - Posted ads click-through rate: 5%
 - Pay per click rate: \$0.05
- Pay per click affiliate program revenue: 60,000 ads x 30 daily views x 30 days x 5% x \$0.05 = \$135,000 per month

Summary (per month)

- Total costs: \$720
- Total profits: \$135,000
- Gain: \$134,280
- Productivity index (Profits/Costs): 187

The bottom line? The more or less masqueraded spam flourishing on social networking sites may seem innocuous at first sight, but again, is very organized and yields outstanding profitability figures.

2.5 What’s next?

Besides being used for fuelling ‘spam 2.0’, social networking site ID theft has plenty of other applications. Just a thought: it is rumoured that 1/3 of the US population alone uses *MySpace*; today record labels, television and movie studios, celebrities and even politicians are leveraging the social network site to promote their identities, projects or causes. What happens if that kind of information falls into the hands of the wrong person? (Remember Paris Hilton’s sidekick?).

At the time of writing, stolen or hijacked social networking site IDs have not been used for distributing malware, to my knowledge. This could, however, very well happen at some point. Let's consider again our hypothetical pool of 6,000 stolen accounts; instead of using them for spraying smart spam on 'friends' comments pages, they could be leveraged as a vector for drive-by-install operations: what if someone resorts to the clickable transparent 'cover-all' image technique described above in 'social worms++' to lead each visitor who'd click anywhere on the page to an MPack server? Within a week, that would easily drive 300,000 individual users to the MPack server (assuming an average of 50 different individual viewers per week per profile). According to the MPack infection statistics, this would in turn result in 36,000 individual successful infections (300,000 x 12%); worse: should Samy's worm have automated this strategy, it would have driven millions of users to the MPack server(s) within a couple of days.

Infected machines could then be used in various business models (see MPack case above) and generate several hundreds of thousands dollars within a short amount of time.

As a matter of course, other popular Web 2.0-like sites are likely to be the target of similar attacks in the near future – *YouTube*, of course, but also *Orkut*, *hi5*, *Facebook*, *blogger*, *SkyBlog*, *Flickr*, to name just a few, are all potential targets (although by and large untouched, at the time of writing).

Purely Web 2.0 sites, such as personal start-pages aimed at syndication (e.g. *NetVibes*), and making a heavy use of AJAX, may be appealing targets as well. Indeed, a simple CSRF flaw could allow attackers to inject sponsored links masquerading as news articles directly in the targeted user's start-page.

As a matter of fact, it does seem that as our data – and in a longer term, the whole desktop – is moving to online applications, threats, more monetized than ever, are following close behind.

3. ADVANCES IN eBAYING

The term 'eBaying' is widely used on fraud-oriented forums and IRC channels. While in most people's minds eBaying just coins the action of legitimately selling and buying goods on *eBay*, cyber criminals use the term exclusively to designate auction site fraud; on that note, very complete 'eBaying guides' are regularly exchanged or sold on IRC.

As a matter of course, eBaying is not new *per se*, but the evolution of fraudsters' strategies over the past two years, both in terms of automation and risk taking, is particularly interesting.

3.1 Plain bogus item

Setting up a fake auction for a non-existent item, cashing the money and disappearing into the shades of cyber space is probably one of the easiest and most direct way to make money on the web illegally. The aforementioned eBaying guides usually come up with extensive guidelines on how to carefully choose your nonexistent items to sell. The key idea is to play on the item's buzz factor, the item's rarity, and to give the potential buyer the feeling that he is getting a real bargain. Combined, those can effectively create some form of excitation or euphoria on the buyer's side, prone to blind him to the extent that he would accept immediate *Western Union*

or *MoneyGram* payment after cancellation of the auction by the generous seller (which, by the way, is another strategy to social engineer the buyer into thinking he is privileged, and doing a real deal).

This fraud scheme has probably been around since the early days of *eBay* itself, and victims have very little chance of getting any of their money back – for one because by accepting 'under the table' immediate payment they breached *eBay* policies, and above all because wire money transfers are mostly anonymous. Indeed, as was detailed in [1], although one should only be able to retrieve the cash involved in a transfer with the MTCN (i.e. the transaction number) and a national ID, in practice this may not always be the case: agencies sitting in third-world countries perform only light ID checking (or no checking at all), for people generally cannot afford national IDs.

However, the fact that in this scheme, the bogus items are most often nonexistent, can give raise to amusing situations. For instance, I remember stumbling across a mixing console that was auctioned with an abnormally low 'buy it now' price. There was a picture of the product, which is shown in Figure 11.



Figure 11: Auctioned mixing console.

I agreed on a price with the seller, and arguing that one can never be too cautious, asked him to provide me with a picture of the console with a pen sitting on it. At first he first refused, pretending that the console was packaged already. But when I ended the negotiation, he came back with the picture shown in Figure 12.

Your sub-average Photoshop job.



Figure 12: *Cough* requested pic *cough*...

3.2 Bogus item with good user feedback

The productivity of the plain bogus item scheme has, however, been tremendously reduced as a result of increasing user awareness (due to awareness campaigns, stories in the media, etc.). Today, few people would buy from a seller whose feedback is thin, let alone plain blank. As a consequence, *eBay* scammers planning to implement a sustainable business based on *eBay* fraud had to refine their strategies, and meet the challenge: how to get a hold of a high-feedback *eBay* account at will?

Two solutions emerged: either steal it or craft it.

3.2.1 Steal it: *eBay* phishing

Since anti-phishing organizations started to publish statistics on most popular phishing targets, *eBay* along with *PayPal* has regularly been at the top of the roster, being targeted by as many as 20 times more phishing emails than the most popular banks [11].

Once an account with a reasonable feedback score (both in terms of positiveness of comments and number of comments) has been hooked, hijacking it (by changing the password and the registered email) and setting up the bogus auction is trivial. The phishing operation itself is a bit trickier to set up, but all the basic bricks needed for that are widely available on specialized IRC channels and fraud-oriented web-forums (see [1] for more details). The following business model can then be devised, as an example:

Costs (covering the actual phishing operation)

- Phishing kit: scam letter + scam page: \$5
- Fresh spam list: \$8
- A fistful of PHP-mailers to spam out 100K emails for 6 hours: \$30
- Hacked site for hosting scam page for a couple of days: \$10
- Valid cc to register domain name: \$10

Profits

- Assuming
 - A phishing success rate of 0.0001 (10 accounts phished for 100K phishing emails sent)
 - Half of the hooked accounts suitable for bogus auction set up (i.e. sufficient feedback)
 - An average price of \$4,000 for the items sold (guitar amps, plasma TV sets, etc.)
- 5 x \$4,000 = \$20,000

Summary

- Total costs: \$63
- Total profits: \$20,000
- Productivity index (Profits/Costs): 317

It is worth noting that should \$20,000 not be the biggest one-shot jackpot ever hit by a cyber criminal, the productivity index here is outstanding – close to what one may obtain with cashing phishing via local drops (see [1]), but with much less risk. And again, it is only one third of the heroin business’ theoretical productivity index (with much, much less risk).

Of course, this is merely an example, and it is always possible to boost the productivity index by attempting to sell highly valued items (i.e. cars, boats, high-profile Hi-Fi set-ups etc.); however in that case, not only do the risks increase exponentially, but there are also obviously fewer potential buyers (and those are also more cautious, given the odds), making the scheme less robust and overall more aleatory.

3.2.2 Craft it: broker bots

Today, numerous *eBay* sellers offer ‘buy it now’ items at the price of 1 cent with no delivery cost (usually *eBooks*, pictures, wallpapers, etc.). Figure 13 depicts the feedback profile of such a seller.

This is just a small excerpt, but the same striking pattern is repeated over pages and pages: most user names are made of six to eight random letters and bear around 15 evaluations. Having a look at these profiles reveals that they have bought roughly the same items – all for 1 cent. Figure 14 draws a comparison of two of such buyers’ profiles.

Again, a sharp eye may notice that feedback comments received from sellers are identical, and read almost in the same order. This is because most 1-cent-plus-no-delivery-cost sellers automate the whole transaction: should someone buy their *eBooks* for one cent each, some scripts email it automatically to the buyer, and leave a standard feedback comment on the buyer’s profile.

Now, if we recollect everything, the following is probably happening:

1. Someone is creating a very large number of randomly named, fake user accounts (probably in a more or less automated fashion).
2. Those fake users, powered by automated web spider software, are set to scavenge *eBay* for 1-cent ‘buy it now’ items and buy them.



Figure 13: Feedback profile a penny-seller.

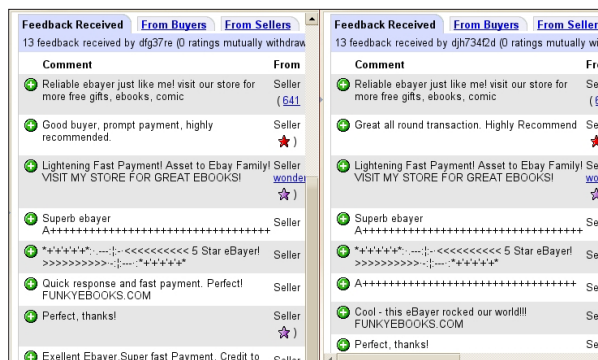


Figure 14: ‘Spot the seven differences’ game, geek-style.

3. Automatically, the 1-cent item seller script is emailing the buyer with the item, and posts its standard feedback on his profile.
4. The fake user automatically responds with a standard feedback comment on the seller's profile.

In a nutshell: two bots are talking. And doing business.

This is a good example of a 'cyber' symbiotic phenomenon (a.k.a. a win-win situation): sellers are making cash without doing anything, and scammers owning the fake accounts are building positive feedback, again, while sleeping, watching porn, or chatting on IRC – and only for a fistful of bucks. Again, let's call numbers in and quantify a business model based on this scheme:

Costs:

- Building 100 accounts with 15 positive feedback messages each: $0.1 \times 100 \times 15 = \15

Profits:

- Assuming:
 - Moderately priced bogus items (about \$100), so that potential victims tend to discard any more advanced security check than a quick look at the feedback page
 - A moderate scam success rate of 1/4 (i.e. one auction out of four will end with victim's effective payment)
- $100 \times 1/4 \times \$100 = \$2,500$

Summary:

- Total costs: \$15
- Total profits: \$2,500
- Gain: \$2,475
- Productivity index (Profits/Costs): 166

Ironically, one of the most popular items among the 1-cent-buy-it-now-with-no-delivery-cost clique is an eBook called 'The Secrets of The 1 Penny Auction'. I do not know what wise advice it features, but one thing is for sure: it includes 'put this eBook on auction for 1 cent'. Anyone willing to waste a penny and report its contents?

3.3 The paid-on-delivery scam: physical action

This is where it becomes really interesting, or should we say worrying. Here is the testimonial of Mr J., a contact of the author, who shall remain anonymous, and who, in his spare time, tracked down a cyber criminal to discover the astonishing scam.

Please do not try this at home.

'Utrecht is the fourth largest city in the Netherlands, and the city centre is one of the most densely populated city centres in the country, making this a perfect spot to place a WiFi-enabled cyber criminal honeypot. The honeypot was able to sniff all 802.11a/b/g packets in about a 200-metre radius, and was programmed to filter out any new connections made to Dutch online marketplaces. Over the course of a few weeks I would correlate this data with reported online crimes and look for suspicious things like connections or groups of connections (grouped either by time or marketplace) that were putting up identical or very similar online ads. After about two weeks, one recurring event caught my attention. A computer would connect to

some open network, try to connect to several IM services like *MSN*, *Gtalk* and *Skype* – but would be cut off before that could connect - client side. The guy probably figured that connecting to his personal IM accounts wasn't such a good idea, but apparently he wasn't smart enough to kill these services before he connected to the wireless networks the next time. Also, there would be set of socks connections (relayed through free socks-proxies) to several free email domains and marketplaces, as well as to the image servers of *Google*.

'Sure enough, this user seemed to be putting up several online ads – with a whole variety of marketplace identities. Curious about what kind of scam he would use – I sniffed his email account passwords and started going through his daily digest of email communication. This guy was a busy busy [guy] indeed. Hundreds of emails a day and from what I could see, on average, there were about two or three users that agreed on delivery by snail-mail every day. Twenty bucks for a DVD or \$2,000 for a brand new *Dell XPS* laptop. He sold it all, all day long!

'I figured that the only way of really checking if this guy was a genuine cyber criminal or if I had just stumbled upon a rare case of mobile online traders in 70 Volkswagen vans, was to check it out. So armed with my laptop, PDA (for fast-paced pursuit) and his MAC address, I went out to find him.

'Sure enough, one day he triggered my Larry Wall script and alerted me that he was in the neighbourhood. *Knock Knock*, the car-window opened 'Hi, my name is J. and it seems that you are misusing my wireless network for your second income, and I'm here for my cut.' I said. A totally astonished and shocked look followed. 'What? Huh? How?' silence followed. I quickly told him that I wasn't a cop, I was not going to report him and didn't want any of his money – just a glance into the why and how. And to my great surprise, he agreed to discuss his story over a beer.

'What followed was a chat over a few beers. Apparently the guy's name was Martin, and Martin learned his scamming craft from an IRC channel where he was recruited to do someone else's bidding. Through an application he got handed by someone else, he did business. He trades 'leads' with points. Points, in turn, could be cashed via *Western Union* or *e-Gold*. A lead consists of a buyer in a certain area, willing to pay a certain amount for a certain product – waiting for the arrival of the product at some date. Other people paid for these leads, and would arrive at the buyer's house, on the expected date, with a package full of turds. Dressed in a genuine *TNT* outfit, they traded they turd-surprise for the amount agreed.'

This leads/points system and the whole IRC recruitment business is not too surprising, and confirms what was demonstrated last year already in [1]: the cyber criminal scene is structured in different layers (buyers, doers, kids, coders, mob, etc.). But what is somewhat stunning here, is that there is a stratum of people who are actually willing to take the risk to show up at your door and deliver you a 'box full of turds' (I guess this was merely a figure of speech from Mr J., by the way) to scam you.

So, what is this? Cybercrime or 'regular' crime? A mix of both? It does seem that while some aspects of our lives are neither totally 'online', nor totally 'real' any more, the same is true for criminality.

CONCLUSION

If cyber criminality is starting to head for new and potentially very juicy grounds such as large, Web 2.0-enabled community sites, one year later most aspects of the conclusion remain unchanged: the tremendously high profitability of cyber criminal schemes, their relative ease of implementation (again, no need to be an elite computer wizard when all the basic bricks are available for purchase on IRC) and the abnormally low risks involved given the odds are absolutely stunning; and undeniably tempting.

Several factors may be invoked to explain such a favourable combination; on top of them is an ever-going issue: the internet is absolutely borderless, while law enforcement – and laws themselves – are strongly tied to states and countries. And governments may not always understand this issue clearly (let alone try to solve it). As a simple example, in 2007, the French presidential elections were held. The final voting round was due to close at 8pm, therefore, although poll institutes all have the results of the vote by 6pm thanks to advanced estimations, it is forbidden by law to publish such results before 8pm. When asked if bloggers should commit to this interdiction, the answer generally was ‘if the blog is in France, yes’. Now how does a blog qualify as ‘in France’? Does that mean the blogger is French? Writes from France? That the blog service has its headquarters in France? That the physical server hosting it is in France? Further, does this statement have any meaning at all? What is the point asking blogs ‘in France’ to commit to this interdiction whereas it is an effortless process for a web user to consult a blog hosted in Singapore or Canada rather than the mouth taped ‘in France’ ones?

This is symptomatic of the current situation when it comes to combating cybercrime: trying to apply state-bound systems of justice to a physically, socially, and culturally borderless entity seems hopeless.

Now, finding an effective solution to that issue – should it be merely theoretical – is a complex task. Having a global and international ‘Internet Department of Justice’ that supersedes local jurisdictions raises endless issues, beginning with the probable refusal of most countries to alienate their justice prerogatives to an international entity. A tighter collaboration between national police forces, forming some sort of Cyber Interpol, sounds like a reasonable solution; however, the police in emerging countries most likely have ‘more important’ things in mind and tend to overlook the cybercrime issue, for it does not produce corpses (which is somehow understandable). And since cybercrime originates, for a consequent part, from emerging countries.

Bonus track: the top 10 most profitable cyber criminal business models

Based on the productivity index, and compiled from both this paper and [1]:

Rank	Business model	P.I. (Profits/Costs)
1	Phishing: cashing via local drops	400 (up to)
2	Bogus auctions from stolen accounts	317
3	Spam 2.0	187
4	Bogus auctions from ‘broker bot’-boosted accounts	166

5	Adware / spyware planting	102 (first month, then grows)
6	Online extortion	32
7	Phished credentials traffic	31 (up to)
8	Mass injections	27
9	Phishing: cashing via offshore accounts	10
10	Carding: ‘Buy Stuff’	9

Figure 15: Cybercrime top 10 profitable activities.

REFERENCES

- [1] http://www.fortiguardcenter.com/reports/dirty_money_on_the_wires.html.
- [2] http://www.wired.com/software/coolapps/news/2007/04/thunderbirdqa_0409.
- [3] <http://en.wikipedia.org/wiki/Xss>.
- [4] <http://www.cgisecurity.com/articles/csrf-faq.shtml>.
- [5] <http://namb.la/popular/tech.html>.
- [6] <http://www.cwire.org/highest-paying-search-terms/>.
- [7] Harris Interactive poll: ‘Friendship in the Age of Social Networking Websites’.
- [8] <http://www.comscore.com/press/release.asp?press=1145>.
- [9] http://forevergeek.com/articles/debunking_the_myspace_myth_of_100_million_users.php.
- [10] <http://namb.la/popular/>.
- [11] <http://www.phishtank.com/stats.php>.