The Impact of Public Information on Phishing Attack and Defense

Tyler MOORE & Richard CLAYTON
Harvard University and the University of Cambridge

Abstract: Attackers compromise web servers in order to host fraudulent content, such as malware and phishing websites. While the techniques used to compromise websites are widely discussed and categorized, analysis of the methods used by attackers to identify targets has remained anecdotal. In this paper, we study the use of search engines to locate potentially vulnerable hosts. We present empirical evidence from the logs of websites used for phishing to demonstrate attackers' widespread use of search terms which seek out susceptible web servers. We establish that at least 18% of website compromises are triggered by these searches. Many websites are repeatedly compromised however the root cause of the vulnerability is not addressed. We find that 17% of phishing websites are recompromised within a year, and the rate of recompromise is much higher if they have been identified through web search. By contrast, other public sources of information about phishing websites actually lower recompromise rates. We find that phishing websites placed onto a public blacklist are recompromised less often than websites only known within closed communities. Consequently, we conclude that strategic disclosure of incident information can actually aid defenders if designed properly.

Key words: security economics, online crime, phishing, transparency.

Information security is of growing interest to policy makers as society becomes more dependent on a reliable communications infrastructure. An economic perspective has proven particularly useful in understanding how attackers and defenders operate and identifying ways to influence their behavior (ANDERSON & MOORE, 2006). A flourishing underground economy has emerged, where profit-motivated criminals exploit the Internet's universal addressability and scale to defraud many unsuspecting citizens (MOORE *et al.*, 2009).

Despite the increase in online criminal activity, information on incidents and the losses caused by such crimes have largely remained hidden from public view. There are several reasons for this. First, firms fear negative publicity which may arise if incidents are openly discussed. Second, some argue that disclosing information on incidents actually aids attackers more than it helps defenders. For example, RANSBOTHAM (2010) has found that

vulnerabilities in open-source software are more frequently exploited by attackers than those present in closed-source software.

Yet there are also clear benefits to public disclosure of security incidents. First, while criminals already know how to operate, the 'good' guys could stand to gain from a fuller understanding of how attacks work. This notion – that open discussion of information liable to abuse is valuable – dates to at least the 17th century (WILKINS, 1641). Second, security economics has identified the lack of reliable information on threats as a key barrier to optimal security investment. Better measurement of the frequency and impact of incidents can help firms better allocate security budgets and provide an incentive to improve behavior. Third, bringing incidents to light can help defenders more quickly identify and plug holes.

So is it better to disclose information on security incidents or keep things hidden? The answer to this question is very timely. Some have called on policy makers to require greater transparency from the private sector. Most US states now require companies that lose personally identifiable information to disclose this fact to affected customers. The European Commission is considering revising the Data Protection Directive to adopt a similar requirement. Furthermore, in a report to the European Network and Information Security Agency (ENISA), ANDERSON *et al.* (2008) called for the collection and publication of comprehensive statistics on losses due to electronic crime, as well as data on the levels of malicious traffic emanating from European Internet Service Providers (ISPs). In a report to the US National Academy of Sciences, MOORE (2010) called for the publication of data on computer infection remediation efforts at ISPs, online-banking fraud losses, and control-system incidents at critical infrastructure operators.

Meanwhile, firms have undertaken a number of collaborative efforts to improve security without disclosing results publicly. Google operates a large blacklist of websites currently infected with malware [1], which allows users to verify whether suspected URLs are malicious without revealing the infected websites. ISPs are tinkering with different ways to fight botnets, but to date, most envision arrangements that keep potentially embarrassing details, such as infection rates and time-to-remediation, hidden from public view.

In this paper, we attempt to shed light on the broader questions surrounding public disclosure of information security incidents by empirically

[1] http://code.google.com/apis/safebrowsing/

examining the particular case of phishing. First, we present evidence that attackers do in fact exploit public information about vulnerable web servers to identify new targets. Second, we compare a large public blacklist of phishing URLs to several private ones, finding that websites appearing in the public list are *less* likely to be recompromised at a later date. This suggests that defenders also take advantage of public information on incidents to reduce the exposure to attacks.

Public information in phishing: targeted web search and URL blacklists

Criminals use web servers to host phishing websites that impersonate financial institutions, to send out email spam, to distribute malware, and for many other illegal activities. To reduce costs, and to avoid being traced, the criminals often compromise legitimate systems to host their sites. Extra files – web pages or applications – are simply uploaded onto a server, exploiting insecurities in its software. Typical techniques involve the exploitation of flaws in the software of web-based forums, photo galleries, shopping cart systems, and blogs. The security 'holes' that are taken advantage of are usually widely known, with corrective patches available, but the website owner has failed to bother to apply them.

The criminals use a number of techniques for finding websites to attack. The most commonly described is the use of scanners – probes from machines controlled by the criminals – that check if a remote site has a particular security vulnerability. Once an insecure machine is located, the criminals upload 'rootkits' to ensure that they can recompromise the machine at will (WATSON *et al.,* 2005). They then exploit the machine for their own purposes, or perhaps sell the access rights on the black market (FRANKLIN *et al.*, 2007). If the access obtained is insufficient to deploy a rootkit, or the criminal does not have the skills for this, the website may just have a few extra pages added, which is quite sufficient for a phishing attack.

An alternative approach to scanners, that will also locate vulnerable websites, is to ask an Internet search engine to perform carefully crafted searches. This leverages the scanning which the search engine has already performed, a technique that was dubbed 'Google hacking' by LONG (2004). He was interested not only in how compromisable systems might be located, but also in broader issues such as the discovery of information that was intended to be kept private. Long called the actual searches 'googledorks',

since many of them rely upon extended features of the Google search language, such as 'inurl' or 'intitle'.

In this paper we examine the evidence for the use of 'evil searches': googledorks explicitly intended to locate machines that can be used in phishing attacks. [2] In the following Section we explain our methodology and detail our datasets. Although it is widely accepted that criminals use these techniques, to our knowledge, this is the first study to document their prevalence 'in the wild'. In the 3rd Section we clearly establish 'cause and effect' between the use of evil searches and the compromise of web servers and estimate the extent of evil searching. In the 4th Section we study website *re*-compromise, showing that over 17% of compromised servers host a phishing website on at least one more occasion. We demonstrate a clear linkage between evil search and these recompromises. However, 'findability' is not always bad. We consider the subset of websites that appear in PhishTank's [3] publicly available list of compromised sites and find evidence that being listed in PhishTank substantially decreases the rate of recompromise, demonstrating the positive value of this data to defenders. Finally, we discuss the difficulties in mitigating the damage done by evil searching, and the limitations on using the same searches for doing good.

■ Data collection methodology

We receive a number of disparate 'feeds' of phishing website URLs. We take a feed from a major brand owner, which consists almost exclusively of URLs for the very large number of websites attacking their company, and another feed that is collated from numerous sources by the Anti-Phishing Working Group (APWG) [4]. We fetch data from the volunteer organization 'PhishTank', which specializes in the URLs of phishing websites. We also receive feeds from two 'brand protection' companies who offer specialist phishing website take-down services. These companies amalgamate feeds from numerous other sources, and combine them with data from proprietary phishing email monitoring systems.

---

[2] While we focus on websites used for phishing, once a site is found it could be used for any malevolent purpose (e.g., malware hosting).

[3] http://www.phishtank.com/

[4] http://www.apwg.org/

Table 1 - Categorization of phishing website hosting, October 2007-March 2008

Type of phishing attack	*Count*	*%*
Compromised web servers	88 102	75.8
Free web hosting	20 164	17.4
Rock-phish domains	4 680	4.0
Fast-flux domains	1 672	1.4
'Ark' domains	1 575	1.4
Total	**116 193**	**100.0**

Although by their nature these feeds have substantial overlaps with each other, in practice each contains a number of URLs that we do not receive from any other source. The result is that we believe that our database of URLs is one of the most comprehensive available, and the overwhelming majority of phishing websites will come to our attention. In principle, we could use capture-recapture analysis to estimate what proportion of sites we were unaware of, as attempted by WEAVER & COLLINS (2007). However, the lack of independence between the various feeds makes a robust estimate of coverage impractical to achieve.

Phishing-website demographics

In this paper we consider the phishing websites that first appeared in our feeds during two periods. Primarily, we examine the six-month period from October 2007 through March 2008. When comparing the public PhishTank list to the private lists, we study phishing websites first reported from October 2007 through November 2008. In both cases, we continued to examine the websites for subsequent recompromise through October 2010.

We can split the identified websites into a number of different categories according to the hosting method used. Table 1 summarizes their prevalence for the six-month sample. By far the most common way to host a phishing website is to compromise a web server and load the fraudulent HTML into a directory under the attacker's control. This method accounts for 75.8% of phishing. It is these sites, and the extent to which they can be located by evil searches, that this paper considers.

A simpler, though less popular approach, is to load the phishing web page onto a 'free' web host, where anyone can register and upload pages. Approximately 17.4% of phishing web pages are hosted on free web space, but since there is no 'compromise' here, merely the signing up for a service, we do not consider these sites any further.

We can also distinguish 'rock-phish' and 'fast-flux' attacks, where the attackers use malware infected machines as proxies to hide the location of their web servers (MOORE & CLAYTON, 2007). A further group, we dub 'Ark', appears to use commercial web hosting systems for their sites. All of these attackers use lengthy URLs containing randomly chosen characters. Since the URLs are treated canonically by the use of 'wildcard' DNS entries, we ignore the specious variations and just record canonical domain names. Collectively, these three methods of attack comprise 6.8% of phishing websites. Once again, because the exploitation does not involve the compromise of legitimate web servers, and hence no evil searching is required, we do not consider these attacks any further.

We observe that some entities reporting phishing websites have handled phishing websites appearing on shared hosting providers in a peculiar way. Many smaller firms operate websites with unique domain names, but host the content on a single server shared by many other websites. One consequence of this arrangement is that poorly-configured hosts will resolve paths on any of the domains hosted on the shared server. For example, a phishing website appearing on the URL http://example1.com/~aardvark/bank.html may also appear on http://example2.com/~aardvark/bank.html if both example1.com and example2.com are hosted on the same server. Some firms report as phishing all URLs for every domain hosted on the shared website, even when the attackers have only transmitted phishing emails referring to one domain. We presume this is done either out of an abundance of caution or to inflate the number of reported phishing websites. In any event, we exclude such duplicates from our analysis.

Website-usage summaries

Many websites make use of The Webalizer [5], a program for summarizing web server log files. It creates reports of how many visitors looked at the website, what times of day they came, the most popular pages on the website, and so forth. It is not uncommon to leave these reports 'world-readable' in a standard location on the server, letting anyone inspect their contents.

[5] http://www.mrunix.net/webalizer/

Table 2 - Evil search terms found in Webalizer logs, June 2007–March 2008.

Search type	*Websites*	*Phrases*	*Visits*
Any evil search	204	456	1 207
Vulnerability search	126	206	582
Compromise search	56	99	265
Shell search	47	151	360

From June 2007 through March 2008, we made a daily check for Webalizer reports on each website appearing in our phishing URL feeds. We recorded the available data – which usually covered activity up to and including the previous day. We continued to collect the reports on a daily basis thereafter, allowing us to build up a picture of the usage of sites that had been compromised and used for hosting phishing websites.

In particular, one of the individual sub-reports that Webalizer creates is a list of search terms that have been used to locate the site. It can learn these if a visitor has visited a search engine, typed in particular search terms and then clicked on one of the search results. The first request made to the site that has been searched for will contain a 'Referrer' header in the HTTP request, and this will contain the terms that were originally searched for.

Types of evil search

In total, over our ten-month study, we obtained web usage logs from 2 486 unique websites where phishing pages had been hosted (2.8% of all compromised websites). Of these usage logs, 1 320 (53%) recorded one or more search terms.

We have split these search terms into groups, using a manual process to determine the reason that the search had been made. Many search terms were entirely innocuous and referred to the legitimate content of the site. We also found that many advanced searches were attempts to locate MP3 audio files or pornography – we took no further interest in these searches. However, 204 of the 1 320 websites had been located one or more times using 'evil' search terms, viz: the searches had no obvious innocent purpose, but were attempts to find machines that might be compromised for some sort of criminal activity. We distinguish three distinct types of evil search and summarize their prevalence in Table 2.

Vulnerability searches are intended to pick out a particular program, or version of a program, which the attacker can subvert. Examples of searches

in this group include 'phpizabi v0.848b c1 hfp1' (CVE-2008-0805 is an unrestricted file upload vulnerability) and 'inurl:com_juser' (CVE-2007-6038 concerns the ability of remote attackers to execute arbitrary PHP code on a server).

Compromise searches are intended to locate existing phishing websites, perhaps particular phishing 'kits' with known weaknesses, or just sites that someone else is able to compromise. Examples include 'allintitle: welcome paypal' and 'inurl:www.paypal.com' which both locate PayPal phishing sites.

Shell searches are intended to locate PHP 'shells'. When attackers compromise a machine they often upload a PHP file that permits them to perform further uploads, or to search the machine for credentials – the file is termed a shell since it permits access to the underlying command interpreter (bash, csh, etc.). The shell is often placed in directories where it becomes visible to search engine crawlers, so we see searches such as 'intitle: "index of" r57.php' which looks for a directory listing that includes the r57 shell, or 'c99shell drwxrwx' which looks for a c99 shell that the search engine has caused to run, resulting in the current directory being indexed – the drwxrwx string being present when directories have global access permissions.

■ Evidence for evil searching

So far, we have observed that some phishing websites are located by the use of dubious search terms. We now provide evidence of evil searches leading directly to website compromise. While difficult to attain absolute certainty, we can show that there is a consistent pattern of the evil searches appearing in the web logs at or before the time of reported compromise.

Linking evil search to website compromise

Figure 1 presents an example timeline of compromises, as reconstructed from our collections of phishing URLs and Webalizer logs. On 30 November 2007, a phishing page was reported on the http://chat2me247.com website with the path /stat/q-mono/pro/www.lloydstsb.co.uk/Lloyds_tsb/logon.ibc.html.

Figure 1 - Screenshot and timeline of a phishing website compromised using evil search

1: 2007-11-30 10:31:33 phishing URL reported: http://chat2me247.com /stat/q-mono/pro/www.lloydstsb.co.uk/lloyds_tsb/logon.ibc.html		
2: 2007-11-30	no evil search term	0 hits
3: 2007-12-01	no evil search term	0 hits
4: 2007-12-02	phpizabi v0.415b r3	1 hit
5: 2007-12-03	phpizabi v0.415b r3	1 hit
6: 2007-12-04 21:14:06 phishing URL reported: http://chat2me247.com /seasalter/www.usbank.com/online_banking/index.html		
7: 2007-12-04	phpizabi v0.415b r3	1 hit

We began collecting daily reports of chat2me247.com's Webalizer logs. Initially, no evil search terms were recorded, but two days later, the website received a visit triggered by the search string 'phpizabi v0.415b r3'. Less than 48 hours after that, another phishing page was reported, with the quite different location of /seasalter/www.usbank.com/online_banking/index.html. Given the short period between search and recompromise, it is very likely that the second compromise was triggered by the search. Also, the use of a completely different part of the directory tree suggests that the second attacker was unaware of the first. Figure 1 shows a screenshot from a web search in April 2008 using the same term: chat2me247.com is the 13th result out of 696 returned by Google, indicating an obvious target for any attacker.

We have observed similar patterns on a number of other websites where evil search terms have been used. In 25 cases where the website is compromised multiple times (as with chat2me247.com) we have fetched Webalizer logs in the days immediately preceding the recompromise (because we were studying the effects of the initial compromise). For these sites we are able to ascertain whether the evil search term appears before compromise, on the same day as the compromise, or sometime afterward.

Figure 2 - Timeline of evil web search terms appearing in Webalizer logs

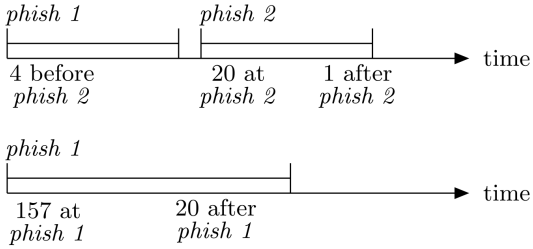


Figure 2 shows a timeline for the 25 websites with Webalizer data before and after a second compromise. For 4 of these websites, the evil search term appeared before the recompromise. For the vast majority (20), the evil search term appeared on the day of the recompromise. In only one case did the evil search term appear only after recompromise. Since most evil terms appear at or before the time of recompromise, this strongly suggests that evil searching is triggering the second compromise. If the evil searches had only occurred after the compromise, then there would have been no connection.

We also examined the Webalizer logs for an additional 177 websites with evil search terms but where the logs only started on, or after, the day of the compromise (see Figure 2). Again, in most cases (157) the evil search term appeared from the time of compromise. Taken together, evil search terms were used at or before website compromise 90% of the time. This is further evidence that evil searching often precedes the compromise of web servers.

Estimating the extent of evil search

We can use phishing websites with Webalizer logs to estimate the overall prevalence of evil search when servers are compromised and used to host phishing websites. Recall that we have obtained search logs for 1 320 phishing websites, and that 204 of these websites include one or more evil search terms in these logs. Frequently, the record shows one visit per evil search. Unfortunately, Webalizer only keeps a record of the top 20 referring search terms. Hence, if a site receives many visitors, any rarely occurring search term will fall outside the top 20. We therefore restrict ourselves to considering just the 1 085 Webalizer-equipped hosts that have low enough traffic so that even search terms with one visit are recorded. Of these hosts, 189 include evil search terms, or approximately 17.6% of the hosts in the sample. Viewed as a sample of all compromised phishing websites, the 95% confidence interval for the true rate of evil searching is (15.3%, 19.8%).

This estimate is only valid if the hosts with Webalizer logs represent a truly random sample. A number of factors may affect its suitability. First, running Webalizer (or programs that it may be bundled with) may affect the likelihood of compromise. We have no evidence for any such effect. Second, sites running Webalizer are not representative of the web server population as a whole. Webalizer typically runs on Unix-like operating systems. Since many compromised servers run on Windows hosts, we cannot directly translate the prevalence of evil web search terms to these other types. Third, evil searches are only recorded in the website logs if the attacker clicks on a search result to visit the site. Using automated tools such as Goolag (Cult of the Dead Cow, 2008), or simple cut and paste operations, hides the search terms. This leads us to underestimate the frequency of evil searches. On balance, we feel sites with Webalizer logs are a fair sample of all websites.

Other evidence for evil searches

There is a substantial amount of circumstantial evidence for the use of evil searches by criminals seeking machines to compromise. Hacker forums regularly contain articles giving 'googledorks', sometimes with further details of how to compromise any sites that are located. However, published evidence of the extent to which this approach has replaced older methods of scanning is hard to find, although the topic is already on the curriculum at one university (LANCOR & WORKMAN, 2007).

LaCour examined a quarter of the URLs in the MarkMonitor phishing URL feed, and was reported (HIGGINS, 2008) as finding that, "75% had been created by using some 750 evil search terms, and the related PHP vulnerabilities". Unfortunately, he was misquoted (LACOUR, 2008). LaCour did collect 750 evil searches from hacker forums, but he did not establish the extent to which these were connected to actual machine compromises.

What LaCour was able to establish from his URL data was that for the October to December 2007 period, 75% of attacks involved machine compromise, 5% were located on free web-hosting and 20% were the categories we have called rock-phish, fast-flux and Ark. These figures are roughly in line with our results in Table 1 above. He then observed, from the paths within URLs, a strong link to PHP vulnerabilities, particularly 'Remote File Inclusion' (RFI) (DAUSIN, 2008). This led him to speculate that evil searches and subsequent RFI attacks may be used in up to 75% of attacks.

■ Phishing website recompromise

Removing phishing websites can be a frustrating task for the banks and other organizations involved in defending against phishing attacks. Not only do new phishing pages appear as fast as old ones are cleared, but the new sites often appear on the web servers that were previously compromised and cleaned up. This occurs whenever the sysadmin removing the offending content only treats the symptoms, without addressing the root problem that enabled the system to be compromised in the first place.

We now provide the first robust data on the *rate* of phishing-website recompromise. Recompromise can serve as a good metric of the effects of public information on both attack and defense. Attackers use evil search terms to discover websites, which could lead to higher recompromise rates. Meanwhile, defenders that identify vulnerable hosts and fix them can lower recompromise rates. In this section we present evidence of how evil search raises the likelihood of recompromise and how public blacklists reduce the incidence of recompromise.

Identifying when a website is recompromised

Websites may be recompromised because the same attacker returns to a machine that they know to be vulnerable. Alternatively, the recompromise may occur because a different attacker finds the machine and independently exploits it using the same vulnerability, or even a second security flaw. We think it unlikely that a single attacker would use multiple security flaws to compromise a machine when just one will do the trick.

The general nature of the security flaw that has been exploited is often quite obvious because the phishing pages have been added within particular parts of the directory structure. For example, when a particular user account is compromised the phishing pages are placed within their file space; when a file upload vulnerability is exploited, the pages are put in sub-directories of the upload repository. However, since it is not always possible to guess what exploit has been used, we instead consider how much time elapses between phishing reports to infer distinct compromises. If two phishing websites are detected on the same server within a day of each other, it is more likely that the same attacker is involved. If, instead, the attacks are months apart, then we believe that is far more likely that the website has been rediscovered by a different attacker. We believe that attackers usually have a relatively small

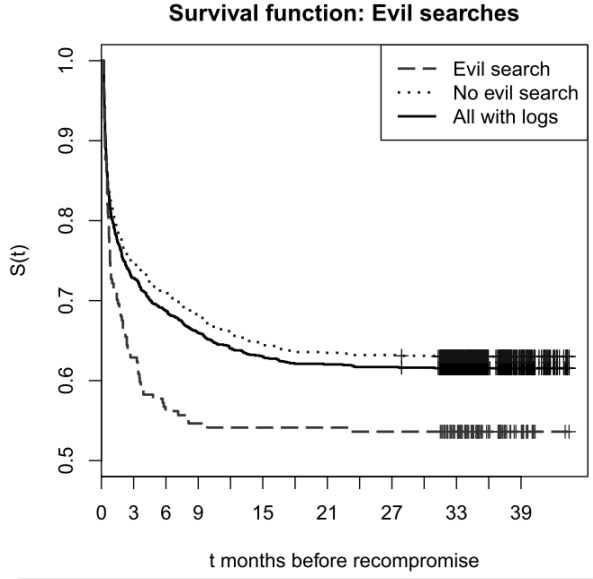
number of machines to exploit at any given moment and are unlikely to keep compromised machines 'for a rainy day' – this is consistent with the short delay that we generally see between detection (evil search logged) and use (phishing website report received).

Our equating of long delays with different attackers is also based on the distribution of recompromises over time. If we treat every phishing site on a particular server as a different attack, whatever the time delay, then we observe a recompromise rate of 20% after 5 weeks, rising to 30% after 24 weeks. If we insist that there is a delay of at least 3 weeks between attacks to consider the event to be a recompromise, then the rates change to 2% after 5 weeks and 15% after 24 weeks. The long term rates of recompromise vary substantially for cut-off points of small numbers of days, which we believe reflects the same attackers coming back to the machine. However, long term rates of recompromise hardly change for cut-off times measured in weeks, which is consistent with all recompromises being new attackers.

An appropriate cut-off point, where there is only a small variation in the results from choosing slightly different values, is to use a gap of one week. We therefore classify a phishing host as recompromised after receiving two reports for the same website that are at least 7 days apart. Using a 7-day window strikes a reasonable balance between ensuring that the compromises are independent without excluding too many potential recompromises from the calculations. As a further sanity check, we note that for 83% of website recompromises occurring after a week or longer, the phishing page is placed in a different directory than previously used. This strongly suggests that different exploits are being applied, and therefore, different attackers are involved.

The rate of website recompromise should only be considered as a function of time. Simply computing the recompromise rate for all phishing websites in the October to March sample would skew the results: websites first compromised on October 1st would have six months to be recompromised, while websites first compromised in late March would have far less time. We handle this in two ways. First, we have continued to check our phishing lists for recompromise through October 2010, so we expect that most recompromises would have occurred by then. However, we cannot state with certainty that a website will never be recompromised. We can only state that it has not yet been recompromised. Fortunately, this is a standard problem in statistics, and we can solve the problem using survival analysis. Websites that have not been recompromised at the end of our study are said to be right-censored.

Figure 3 - Survival analysis shows that websites with evil searches in their logs are recompromised faster and more often than websites without evil searches in their logs



Cox-proportional hazard model
$h(t) = \exp(\alpha + \text{HasEvilSearch}x_1)$

	coef.	exp(coef.)	Std. Err.)	Significance
HasEvilSearch	0.32	1.38	0.12	$p < 0.001$

log-rank test: Q=7.75, $p < 0.01$

Evil searching and recompromise

We established that evil searches can precede website compromise. We now show that the evil searches are linked to much higher rates of recompromise. Figure 3 compares the recompromise rates for hosts in the Webalizer sample. The survival function *S*(*t*) measures the probability that the time between compromises is greater than time *t*. The survival function is similar to a complementary cumulative distribution function, except that the probabilities must be estimated by taking into account censored data points. We use the standard Kaplan-Meier estimator (KAPLAN & MEIER, 1958) to estimate the survival function for recompromise durations, as indicated by the solid line in Figure 3. For instance, *S*(1) = 0.801, which means that 19.9% of websites with search logs are recompromised within one month of the first compromise. The median time before recompromise is undefined, since over half of the phishing websites are not recompromised.

Also noteworthy is that at the maximum time, $S(\max) = 0.615$. Empirical survival estimators such as Kaplan-Meier do not extrapolate the survival distribution beyond the longest observed lifetime, which is just over three years in our sample. What we can discern, nonetheless, is that 61.5% of websites are not recompromised during the span of our investigation.

Figure 3 also plots the survival functions for websites where evil search terms are present in the server logs (dashed line) and where evil search terms are not found (dotted line). Note that $S_{evil}(1) = 0.722$, while $S_{no\ evil}(1) = 0.816$. In other words, 18.4% of websites that have likely not been discovered by evil search are recompromised within one month, compared to 27.8% of websites that have been discovered by evil search. The gap between websites found through evil search and others grows as more time is allowed for recompromise: 37% of websites found through evil search are recompromised within three months, compared to 25% for websites without evil search. After a year, 46% of websites with evil search terms are recompromised, compared to 34% for websites without such terms. We conclude that vulnerable websites found via web search may be repeatedly rediscovered and recompromised until they are finally cleaned up.

But are these differences statistically significant? To compare the recompromise rates of websites with evil search terms in the logs to websites lacking such terms in the logs, we use a Cox proportional hazard model (COX, 1972) of the form:

$$h(t) = \exp(\alpha + \mathrm{HasEvilSearch}x_1)$$

Note that the dependent variable included in the Cox model is the hazard function. The hazard function $h(t)$ expresses the instantaneous risk of 'death' at time t, where in our context 'death' means recompromise. Cox models are used on survival data instead of standard regression models, but the aim is the same as for regression: to measure the effect of different independent factors (in our case, the existence of evil search terms in the server logs) on a dependent variable (in our case, time to recompromise).

The results are presented in the table in Figure 3. The model finds the presence of evil search terms in the server logs to be significantly correlated with shorter time to recompromise. Interpreting the coefficients in Cox models is not quite as straightforward as in standard linear regression; exponentiated coefficients (column 3 in the table) offer the clearest interpretation. The value exp(HasEvilSearch)=1.38 indicates that the presence of evil search terms increases the hazard rate by 38%.

PhishTank and recompromise

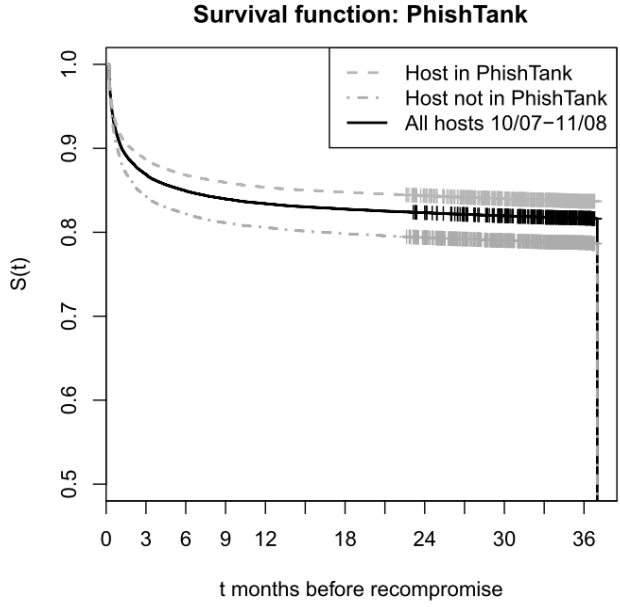
We have shown that attackers use web search to find websites to compromise. We now consider whether they are using public phishing website blacklists as an alternative way to find sites to compromise, or instead if the public nature of the blacklist helps defenders remediate infected hosts.

Phishing-website blacklists provide valuable data for 'phishing toolbars' that block visits to fraudulent websites, and are used by take-down companies to identify websites to remediate. Most blacklists are kept hidden: Google's Safe Browsing API only allows users to verify suspected URLs, while the APWG's blacklist is only available to members. One argument for keeping the lists private is that attackers might mine the lists to identify new targets for recompromise. Another rather different argument is made by the take-down companies who fear that publishing helps competitors to free-ride off their hard work in compiling the lists.

In contrast, PhishTank provides an open source blacklist which is generated and maintained through web-based participation. Users are invited to submit URLs of suspected phishing websites and verify each other's entries. PhishTank argues that by making its blacklist available free of charge, consumers are better protected since more support staff at ISPs and sysadmins are informed of compromised websites in need of cleanup. Other companies sell phishing feeds to aid ISPs in this manner, but PhishTank's free service may be more widely adopted. Furthermore, PhishTank has explicitly designed its blacklist to be helpful to defenders (especially ISP 'abuse teams'), adding features such as searches for phishing sites based on ASNs and RSS feeds of new entries within an ASN.

We set out to empirically test whether publicizing phishing websites helps or harms the cause of defenders, using the recompromise rate as a metric. It is unfair to simply compare recompromise rates for sites PhishTank knows about with those of which it is unaware. While aiming to be comprehensive, in practice PhishTank fails in this aim, and is aware of only 59% of the phishing websites in our collection. Since some of our other URL feeds get some of their data from PhishTank, it is more accurate to view PhishTank as a subset of the phishing URLs we record. So although PhishTank has a roughly even chance of recording a particular phishing incident, there will be further chances to record the host if it is recompromised. This biases PhishTank's record to include a disproportionate number of hosts where multiple compromises occur.

Figure 4 - Survival analysis shows that phishing websites in PhishTank's public blacklist are recompromised less often than phishing websites that only appeared in a private list



Cox-proportional hazard model $h(t) = \exp(\alpha + \text{InPhishTank}x_1)$				
	coef.	exp(coef.)	Std. Err.)	Significance
InPhishTank	−0.29	0.75	0.014	$p < 0.001$
log-rank test: Q=397.4, $p < 0.001$				

Consequently, we apply a fairer test to determine whether a host's appearance in PhishTank makes it more likely to be recompromised. We compare the recompromise rates of new hosts following their first compromise. 100 735 hosts were compromised and used in phishing attacks between October 2007 and November 2008. 59 593 hosts detected by PhishTank during their first reported compromise are compared against 41 142 hosts missed by PhishTank during the first compromise. Because we are interested in measuring the impact of publication in PhishTank, we exclude any hosts that first appeared in PhishTank prior to October 2007.

The results presented in Figure 4 show that new websites appearing in PhishTank are consistently *less* likely to be recompromised than new websites that do not appear in PhishTank. Within one month, PhishTank-aware phishing websites are recompromised 8% of the time, compared to 11% for sites not reported to PhishTank.

A similar trend holds for recompromised websites within 3 months, with recompromise rates around 11% for websites known to PhishTank, compared to 16% for websites not appearing in PhishTank's public list. A roughly five percentage point difference in recompromise rates for websites appearing in PhishTank compared to sites that remain hidden persists as the time allowed for recompromise extends to three years.

Using a Cox proportional hazard model similar to that described above, we again find these differences to be highly statistically significant. The value exp(InPhishTank) =0.75 indicates that publishing phishing websites in PhishTank corresponds to a 25% reduction in the hazard rate.

The black solid line in Figure 4 provides a robust measure of the overall recompromise rate of phishing websites during the 14-month sample. 9% of websites are recompromised within one month of the initial compromise, rising to 13% within 3 months and 17% within one year.

We note that the overall recompromise rate observed here is substantially lower than the recompromise rate observed in Figure 3 when examining only the 1 085 websites where we have access to the Webalizer logs. What might explain the discrepancy in the recompromise rates for the Webalizer sample? One factor is that the sites with Webalizer logs, by definition, were accessible at least once shortly after being reported. This is not the case for all hosts – some phishing websites are completely removed before we are able to access them. [6] Given that the survival function in Figure 4 is based on all 100 000 hosts observed in 14 months, we expect that this measure of overall website recompromise is more reliable.

Based on our data analysis, we conclude that the good offered by PhishTank (better information for defenders) currently outweighs the bad (exploitation of compromised websites by attackers). However, the use of PhishTank by both attackers and defenders might change dynamically over time. Consequently, we believe that continued monitoring is necessary in case attackers begin to leverage PhishTank's public blacklist.

[6] Many sites that are compromised are long-abandoned blogs and image galleries. It is not surprising that a number of these are removed altogether, rather than being cleaned up and left publicly available.

■ Mitigation strategies

Thus far we have demonstrated clear evidence that evil searches are actively used to locate web servers for hosting phishing websites. We have also shown that server re-compromise is often triggered by evil search. Therefore, we now consider how evil searches might be thwarted, in order to make the criminals' task harder. We set out and review a number of mitigation strategies, the first two of which can be implemented locally, whereas the others require action by outside parties. Unfortunately each has drawbacks.

Strategy 1: Obfuscating targeted details

Evil searches could be made less effective if identifying information such as version numbers were removed from web server applications. While this might make it a bit harder for attackers to discover vulnerable websites, it does nothing to secure them.

DAMRON (2003) argued for obfuscation by noting that removing the version numbers from applications is easy for the defender, while adding a significant burden for the attacker. However, defenders also stand to gain from detailed application information, as the presence of a version number can assist sysadmins in keeping track of which of their users continues to run out of date software.

We note that very few of the evil search terms we examined contained explicit version numbers, but merely sought to identify particular programs. The final objection to this obfuscation strategy is that obscuring version numbers still leaves users exposed to 'shotgun' attackers who run all of their exploits against every candidate site without worrying whether or not it is running a vulnerable version.

Strategy 2: Evil search penetration testing

Motivated defenders could run evil searches to locate sites that might be compromised and then warn their owners of the risk they were running. For many evil searches, which only return a handful of exploitable sites amongst many thousands of results, this is unlikely to be an effective scheme. Furthermore, the search results are usually just hints that only indicate the potential for compromise. Confirming suspicions normally requires an active attack, which would be illegal in most jurisdictions.

Strategy 3: Blocking evil search queries

An alternative approach is for the search engines to detect evil searches and suppress the results, or only provide links to law enforcement sites. Given their inherent specificity, constructing a comprehensive and up-to-date blacklist of evil searches is likely to be difficult and costly. Blocking some of the more obvious terms (e.g., those found in Long's popular database [7]) is unlikely to be effective if the terms used by the criminals rapidly evolve. In any event, the search engines are unlikely to have any meaningful incentive to develop and deploy such a list.

Strategy 4: Removing known phishing sites from search results

The low-cost option of removing currently active phishing sites from the search results has almost nothing to recommend it. Search engines suppress results for known child-pornography sites, and Google prevents users from clicking through to sites that are hosting malware (DAY *et al.*, 2008) until they are cleaned up (MAVROMMATIS, 2007). However, phishing presents different circumstances. Malware is placed on high traffic sites where removal from search results is a powerful incentive towards getting it removed, but phishing sites are often on semi-abandoned low traffic sites where the incentive to remove will be limited. Although the evil search will not work while the phishing site is active, the site will be findable again as soon as the fraudulent pages are removed. This approach would also prevent any use of searches by defenders, which means that it does some harm as well as doing little good.

Strategy 5: Lower the reputation of previously phished hosts discoverable by evil search terms

In addition to flagging active phishing URLs, website reputation services such as SiteAdvisor [8] already give a warning for websites that consistently host malicious content. Since we have shown that a substantial proportion of systems that host a phishing website are later recompromised, such services might mark previously compromised hosts as risky. Furthermore, it would be entirely prudent to proactively flag as a much higher risk any hosts used for phishing which can also be found by evil search terms. The

7 http://johnny.ihackstuff.com/ghdb.php

8 http://www.siteadvisor.com/

magnitude of the risk should reflect our finding that about half of these sites will be recompromised within a year of the first compromise.

In contrast to the difficulties in countering evil search, we are optimistic that the use of public blacklists can help defenders in the fight against phishing and beyond. Security firms' refusal to share data on incidents brings with it significant societal costs. For phishing, a refusal to share between take-down firms has greatly slowed down the speed of website removal and increased consumer exposure to attacks (MOORE *et al.* 2009). Openly publishing data on security incidents promises to increase the efficiency of defense at low cost. Unfortunately, the competitive interests of the information security industry may preclude closer cooperation in this manner, so guidance from policy makers may be required.

■ Related work

As indicated earlier, very little academic research has examined the use of search engines to compromise websites. However, researchers have recently begun to recognize the importance of empirically studying electronic crime. THOMAS & MARTIN (2006) and FRANKLIN *et al.* (2007) have characterized the underground economy by monitoring the advertisements of criminals on IRC chatrooms. PROVOS *et al.* (2008) tracked malicious URLs advertising malware, finding that 1.3% of incoming Google search queries returned links to malware-distributing URLs. MOORE & CLAYTON (2007) studied the effectiveness of phishing-website removal by recording site lifetimes. COLLINS *et al.* (2007) used NetFlow data on scanning, spamming and botnet activity to classify unsafe IP address ranges. WARDMAN *et al.* studied common strings in phishing URLs and identified the underlying vulnerabilities (2009). The current work contributes to this literature by measuring the prevalence of evil search terms for compromising websites and the impact on site recompromise.

Another related area of literature is the economics of information security (ANDERSON & MOORE, 2006). One key economic challenge identified by this literature is overcoming asymmetric information. Better measurement of security is needed, from the prevalence of vulnerabilities in competing software to the responsiveness of ISPs in cleaning up infected hosts. Publishing accurate data on website recompromise can identify serial underperformers and highlight opportunities for improvement. Google and

StopBadware [9] publicly disclose infected websites in search resutls, and it has been claimed that this disclosure encourages prompt cleanup (DAY *et al*., 2008). At a policy level, ANDERSON *et al.* (2008) have recommended that regulators collect better data on system compromise and use it to punish unresponsive ISPs.

■ Conclusion

In this paper, we have presented clear evidence that the criminals who are compromising web servers to host phishing websites are using Internet search engines to locate vulnerable machines. We have found direct evidence of these 'evil searches' in 18% of our collection of Webalizer logs from phishing sites, and believe the true prevalence to be even higher.

We have also shown a clear linkage with the recompromise of servers. The general population of phishing websites exhibits a recompromise rate of 17% after one year, but where evil searches are found in the logs, the rate reaches 46%. Although the use of evil searches has been known about anecdotally, this is the first paper to show how prevalent the technique has become, and to report upon the substantial rates of recompromise that currently occur.

In contrast, phishing website URLs that are made public by the PhishTank database currently enjoy a statistically significant reduction in their recompromise rates. This suggests that defenders are able to use information gleaned from the database in order to reduce criminal attacks, and that the sometimes touted benefits of keeping attack data hidden from public view may in fact be harmful.

Other strategies for mitigating evil search that work by limiting attackers' access to information – obfuscating version numbers, filtering search results, blocking evil search queries – we also consider to be flawed. The most promising countermeasure we discuss is to incorporate a website's likelihood of recompromise into the calculation of its reputation. More broadly, our research suggests that policy makers should encourage the publication of more information that can help the Internet's defenders fix problems as they arise.

9 http://www.stopbadware.org/

References

ANDERSON R., BOEHME R., CLAYTON R. & MOORE T. (2008): *Security Economics and the Internal Market*, European Network and Information Security Agency.

ANDERSON R. & MOORE T. (2006): "The Economics of Information Security", *Science*, 314(5799), pp. 610–613.

COLLINS M.P., SHIMEALL T.J., FABER S., JANIES J., WEAVER R., DE SHON M. & KADANE J. (2007): *Using uncleanliness to predict future botnet addresses*, ACM SIGCOMM Conference on Internet Measurement (IMC), pp. 93–104.

COX D.R. (1972): "Regression models and life-tables", *Journal of the Royal Statistics Society*, Series B, 34, pp. 187–220.

CULT OF THE DEAD COW (2008): *Goolag Scanner Specifications*. http://goolag.org/specifications.html

DAMRON J. (2003): *Identifiable fingerprints in network applications*, USENIX; login, 28(6), pp. 16–20.

DAUSIN M. (2008): *PHP File Include Attacks,* Tipping Point. http://dvlabs.tippingpoint.com/blog/2008/02

DAY O., PALMEN B. & GREENSTADT R. (2008): "Reinterpreting the disclosure debate for web infections", in: M.E. Johnson (Ed.), *Managing Information Risk and the Economics of Security*, pp. 179-197, Springer.

FRANKLIN J., PAXSON V., PERRIG A. & SAVAGE S. (2007): "An inquiry into the nature and causes of the wealth of Internet miscreants", 14th ACM Conference on Computer and Communications Security (CCS), pp. 375–388.

HIGGINS K.J. (2008): *Phishers Enlist Google 'Dorks'*, DarkReading. http://www.darkreading.com/document.asp?doc_id=149324

KAPLAN E.L. & MEIER P. (1958): "Nonparametric estimation from incomplete observations", *Journal of the American Statistical Association*, 53, pp. 457–481.

LACOUR J. (2008): Personal communication.

LANCOR L. & WORKMAN R. (2007): "Using Google hacking to enhance defense strategies", 38th SIGCSE Technical Symposium on Computer Science Education, pp. 491–495.

LONG J. (2004): *Google Hacking Mini-Guide*, informIT. http://www.informit.com/articles/article.aspx?p=170880

MAVROMMATIS P. (2007): *Malware Reviews via Webmaster Tools*. http://googlewebmastercentral.blogspot.com/2007/08/malware-reviews-via-webmaster-tools.html

MOORE T. (2010): "The Economics of Cybersecurity: Principles and Policy Options", *International Journal of Critical Infrastructure Protection*, 3(3-4), pp. 103–117.

MOORE T. & CLAYTON R. (2007): "Examining the impact of website take-down on phishing", 2nd Anti-Phishing Working Group eCrime Researcher's Summit (APWG eCrime), pp. 1–13.

MOORE T., CLAYTON R. & ANDERSON R. (2009): "The Economics of Online Crime*", Journal of Economic Perspectives*, 23(3), pp. 3–20.

PROVOS N., MAVROMMATIS P., RAJAB M. & MONROSE F. (2008): "All your iFrames point to us", 17th USENIX Security Symposium, pp. 1–15.

RANSBOTHAM S. (2010): "An Empirical Analysis of Exploitation Attempts based on Vulnerabilities in Open Source Software, 9th Workshop on the Economics of Information Security (WEIS).

THOMAS R. & MARTIN J. (2006): "The underground economy: priceless", USENIX; login, 31(6), pp. 7–16.

WARDMAN B., SHUKLA G. & WARNER G. (2009): *Identifying Vulnerable Websites by Analysis of Common Strings in Phishing URLs*, 4th Anti-Phishing Working Group eCrime Researchers Summit.

WATSON D., HOLZ T. & MUELLER S. (2005): *Know your Enemy: Phishing.* The Honeynet Project & Research Alliance, http://www.honeynet.org/papers/phishing/

WEAVER R. & COLLINS M.P. (2007): *Fishing for phishes: applying capture-recapture methods to estimate phishing populations*, 2nd Anti-Phishing Working Group eCrime Researcher's Summit (APWG eCrime), pp. 14–25.

WILKINS J. (1641): *Mercury: Or the Secret and Swift Messenger*, Maynard and Wilkins, London.