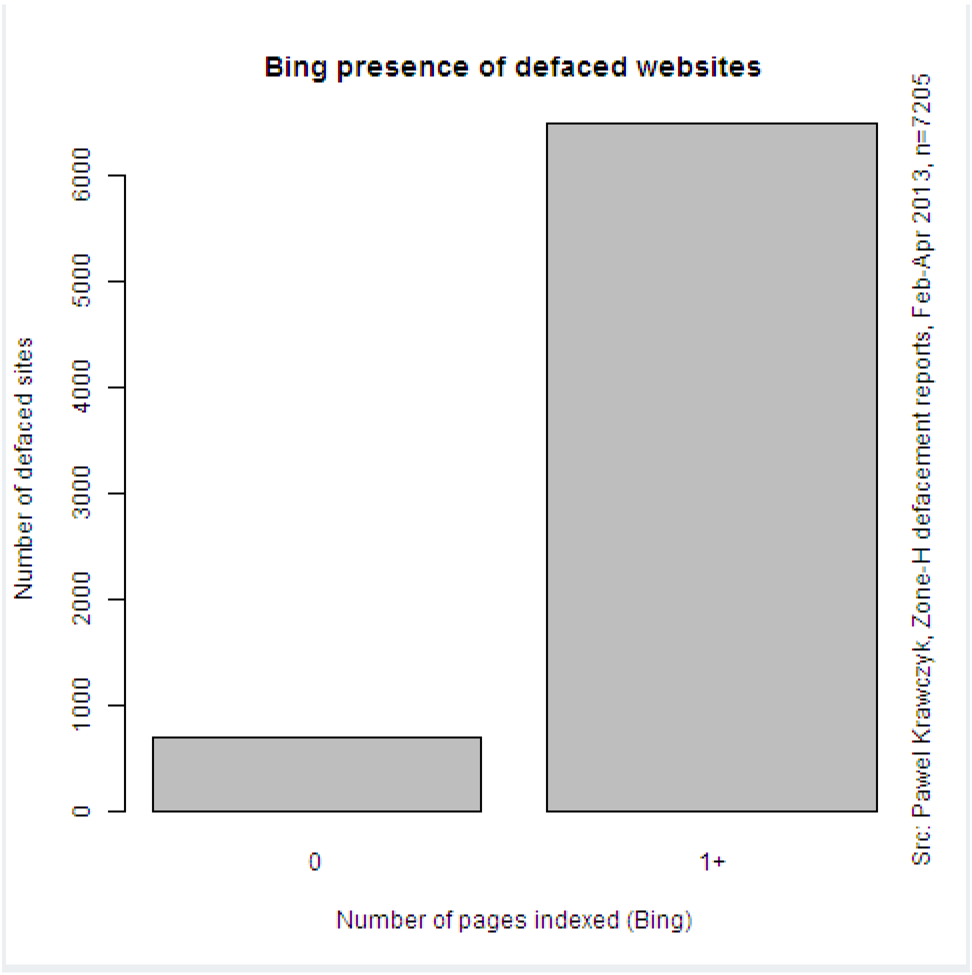
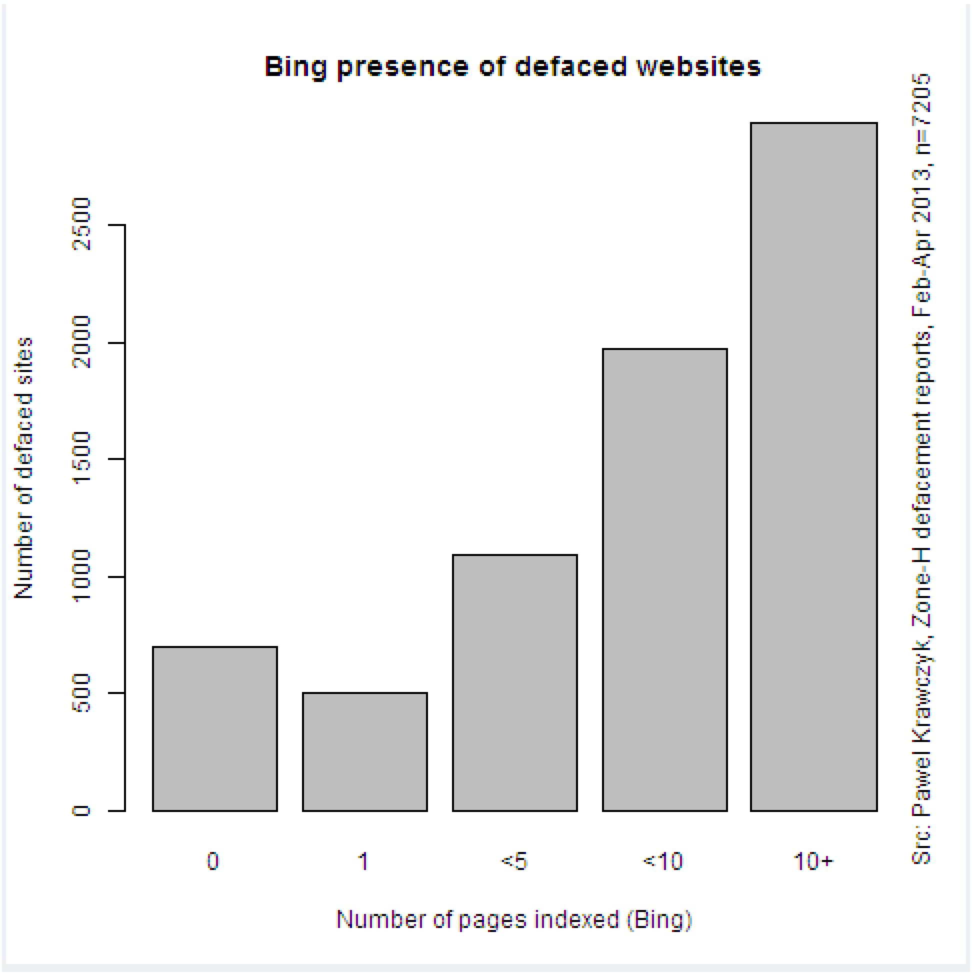
Impact of search engines on opportunistic hacking

Pawel Krawczyk, pawel.krawczyk@owasp.org, Open Web Application Security Project (OWASP)

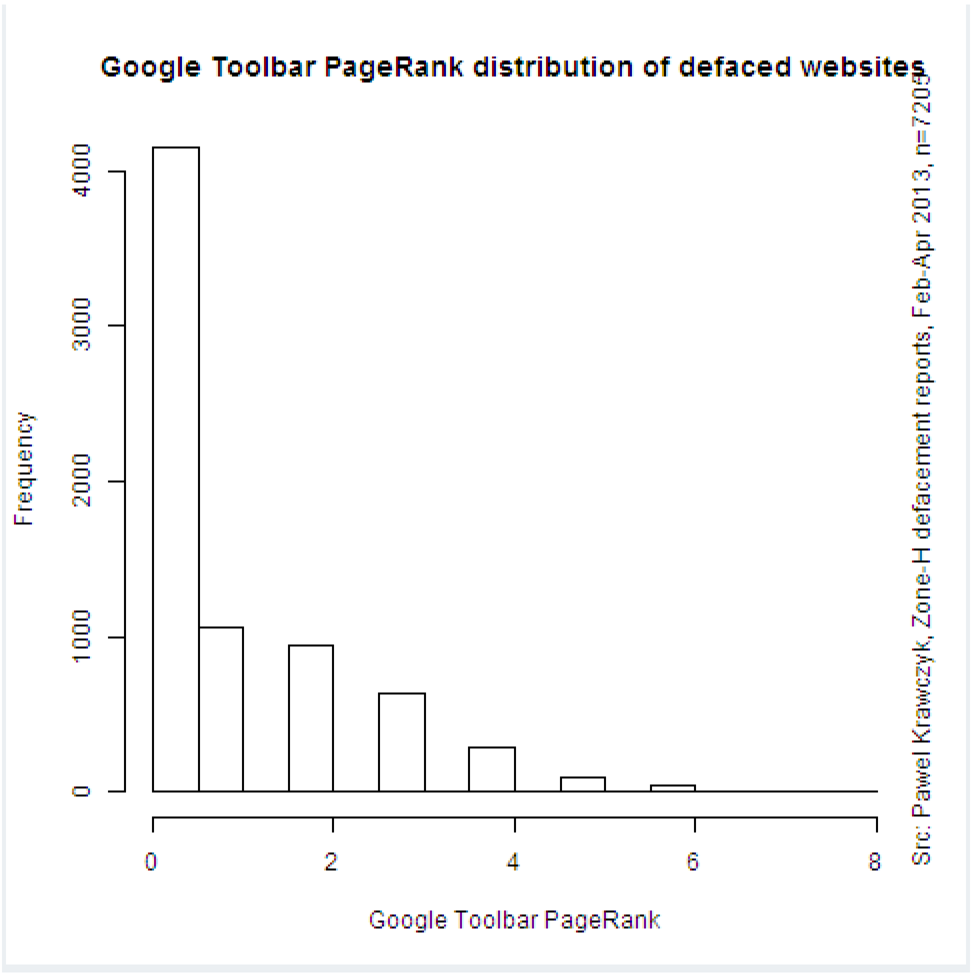
Most website owners struggle to get their website best positioned in search engine results and the rule of thumb seems to be “the more indexing the better”. How does this impact website susceptibility to hacking?
In my penetration testing practice I’ve seen many websites with vulnerabilities visible on the first look that worked for many years on public network and had no signs of past compromise. They could have been hacked at any time — if anyone ever found them and tried. But they were private websites whose URL was directly passed to clients and never linked from any public sites. Search engines just did not have any chance to reach them, but if they did, they could change the situation dramatically. How can we turn this unintended consequence into a conscious advantage?
This chart shows number of pages indexed by Bing on websites that were compromised and announced on Zone-H. Majority of the compromised websites were pretty well indexed by Bing:

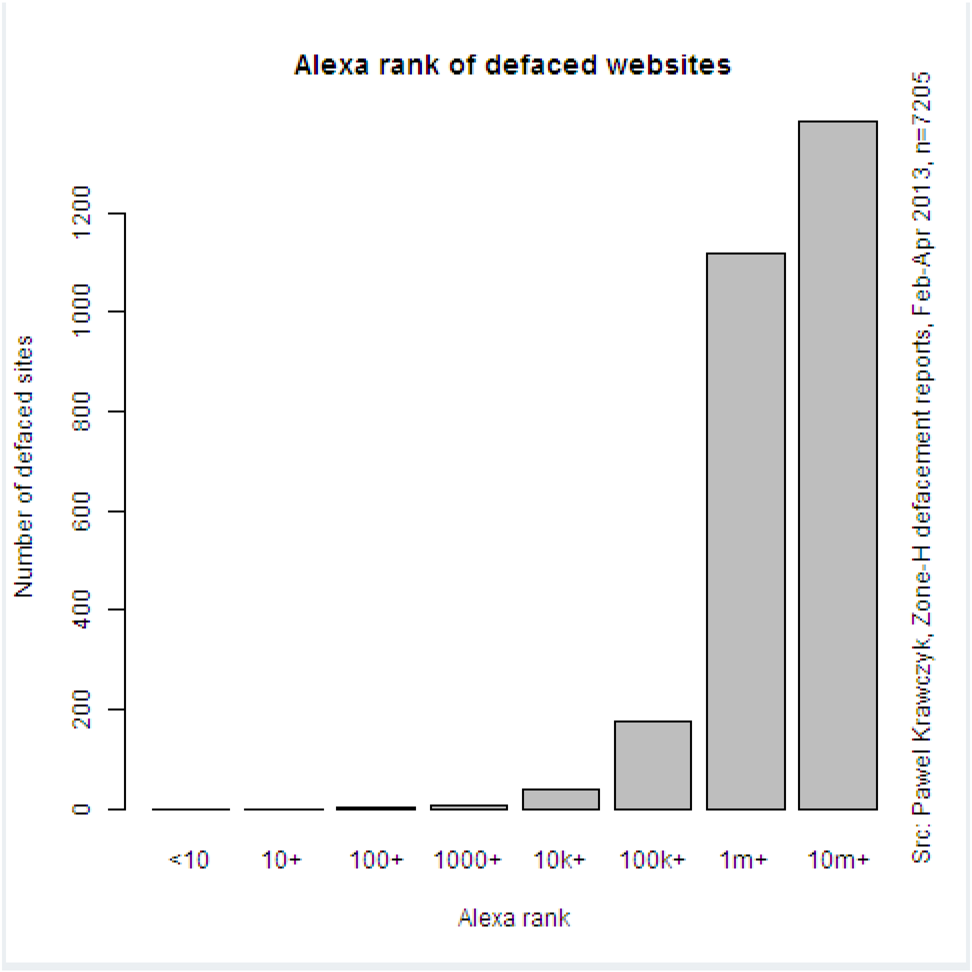


Looking at the same data in more details, we see that the hacked pages were pretty well indexed by search engines. And it was not sufficient that Bing just touched the main page — the better a site was indexed, the more likely it was it gets hacked:



It’s not all that simple however. The following charts show distribution of Google Toolbar PageRank and Alexa Rankfor defaced websites:


Google Toolbar PageRank distribution of defaced websites
Frequency
0
1000
2000
3000
4000
0
2
4
6
8
Google Toolbar PageRank
Src: Pawel Krawczyk, Zone-H defacement reports, Feb-Apr 2013, n=7205



Google rank is from 0 to 10, where zero is lowest popularity and 10 is highest. Alexa classifies websites by their position on their popularity rank, so 1 is the most popular (currently occupied by Facebook). Both charts thus show that it werewebsites of little or medium popularity that were mostly defaced. This can be explained by the fact that popular websites with generate revenue and their owners are more determined to invest in their security. Their risk profile gradually changes from opportunistic hacking to targeted hacking, which was not part of this research.
It is also important to note that hackers do not really need to use search engines to find victims. Most popular vulnerabilities in the wild are found using direct scanning on SSH, SMB, DNS and other ports. Known web application vulnerabilities are also frequently probed by direct connections to the HTTP port. This is however time consuming technique and there are a lot of competing scanners out there, so from attacker's point of view this might be less attractive method compared with search engines.

Methodology

Between February and April 2013 I was scraping Zone-H Archive for new entries. In total, I've collected over 20'000 reports on recently defaced websites. Out of that I've skipped results that were marked as "mass defacements", that is when compromise of single server resulted in defacement of hundreds or thousands of domains hosted there. Eventually I've ended up with a

set of over 7000 defaced websites.
For each defaced website that was newly posted on Zone-H I immediately checked how many results would Bing return for its domain. I was using paid Azure Datamarket Bing API. Fast checking was important, since the very fact of defacement could draw attention to that website and increase search engine coverage.
Initially I have tried to use Google, but there's interesting inconsitency between results returned by consumer Google page and Google Custom Search Engine, which is currently the only legal way to programatically fetch search results from Google. Specifically CSE frequently returns zero results for many domains that have non-zero (or even significanly larger) number of results when checked manually. Bing did not have this problem.
Google Toolbar PageRank was collected using publicly available algorithm and public Google Toolbar servers. Alexa rank was collected using paid Amazon AWIS.

Conclusions

It is well known that botnets do use search engines actively for locating their victims. Botnets usually try to keep low profile and do not announce their victims in public. The data I was looking at came from defacements archive, a kind of hall of fame where hackers compete on number of compromised websites.
The results suggest that proper search engine control on websites is important part of limiting attack surface, just like closing ports on servers and other recognized hardening techniques. At the same time search engine control is a safeguard that is frequently underestimated from both security and business point of view. I've seen many B2B services that revealed all their customer base through permissive search engine controls.
At the same time it's difficult to find a safeguard that is cheaper and easier to set up. Robots.txt, Google and Bing Webmaster Tools are trivial to set up and, most importantly, require minimum or no changes in the application code or templates. This is often the easiest and fastest way to limit exposure of ancient, legacy websites that are still needed, but no longer maintained.

Recommendations

These recommendations based on the following assumptions:

- Most business websites can be classified either as **informational** or **transactional**. The informational sites are open to general public and we definitely do want them in search engines as much as possible (marketing, SEO). The transactional sites contain user specific business data and usually require user login. They usually contain complex business logic and are most susceptible to programming errors and we definitely should limit their visibility in search results.
- Many business websites are intended for limited audience only (often called "extranets"). An anonymous user would just see a login page and nothing else. An example would be a corporate web mail site or electronic banking site for users. We definitely want search engines to stay away from these sites.
- Even if single domain contains both informational and transactional parts, the robots exclusion standard (robots.txt)still offers enough granularity to allow the informational parts, and forbid the transactional ones.

So how robots.txt and webmaster tools can be used for security purposes?

- Prohibit indexing of “extranet” and transactional websites completely. On each such domain set up robots.txt file with these two lines:

```
User-agent: *
Disallow: /
```

- In addition to robots.txt use Google Webmaster Tools and Bing Webmaster Tools for fine grained indexing control on these major search engines.
- Prohibit indexing of transactional parts of websites and selectively allow informational parts. On websites with mixed content set up the following robots.txt (the example assumes that only /login is transactional and remaining pages are informational):

```
User-agent: *
Disallow: /login
```

It is important to understand that search engine control will have absolutely no impact on determined hackers, targeting a particular website of their choice. Web application security is however game of probabilities. Having your transactional website fully indexed by a search engine increased likelihood that it will draw attention of opportunistic hackers or botnets.
And, of course, in an ideal world your website should just not have any exploitable vulnerabilities.

References

- *“Robots exclusion standard”*, Web Robot Pages, 2007
- *“Search engine discovery/Reconnaissance (OWASP-IG-002)”*, OWASP Testing Guide, 2008