MANIPULATING THE INTERNET

Dr. Igor G. Muttik
McAfee AVERT, Aylesbury, UK

Email mig@mcafee.com

ABSTRACT

Traditionally, viruses and other malware were distributed using push techniques – viruses directly or malware authors actively distributed copies around. With the exception of auto-executing worms this method of distribution requires user intervention – a user has to click on an email attachment or launch a program. And users have been told for years to be very cautious about all unsolicited emails. So, in such situations users' defences are higher and such objects are more likely to be avoided or treated with caution.

The situation changes if a user himself is browsing the Internet looking for something. Being motivated to complete what he perceives to be his own task, (s)he is very likely to lower his defences. We are seeing now that 'bad guys' are manipulating the Internet to make sure their malicious software is executed by a large number of unsuspecting users.

So far we have observed at least five different kinds of attack: manipulation of search engines, DNS poisoning, hacking into websites, domain hijacking and exploiting common user mistakes (typos).

We analyse and dissect a case where malicious code was distributed using a technique we called 'index hijacking' – when popular search engines point unsuspecting users to malicious sites. We also investigate a case of 'link hijacking', where a legitimate website pointed users to a bad site involved in 'index hijacking'.

We also discuss DNS poisoning, when users type a URL correctly, but manipulated DNS servers bring them to a completely different location. And finally, we touch on the topic of 'typosquatting' for malware distribution – exploitation of common users' mistakes such as typos in a website's URL.

Important note: many URLs given in this paper point to malicious websites. Do not follow these links. If you do, it is at your own risk.

NEW MALWARE AVENUES

The abundance of websites has turned the Internet into a multicoloured and attractive media where people can get information, exchange views, do their shopping and banking. People were as excited 10 years ago about email as they are excited now with the Internet. Mass-mailing viruses and spam hit email so hard that users' trust in this communication vehicle has suffered very seriously. At the same time malware writing became commercialized. It certainly looks like the traditional malware delivery mechanisms (mass-mailing worms, posting them to newsgroups or spamming URLs to malware) are getting less and less successful.

Naturally, bad guys are looking into new ways of continuing with their nasty business. We are seeing a lot of activity based on instant messaging (new and prolific families have appeared – W32/Kelvir, W32/Bropia, W32/Opanki). Also many new worms and Trojans have appeared for mobile devices (SymbOS/Cabir, SymbOS/CommWarrior, SymbOS/Skulls families). But perhaps the most important shift is that the bad guys are very seriously turning their resources to exploiting the backbone of the Internet – the web. This vehicle is going to stay with us for a long time and, thus, should give the highest return on investment for the bad guys.

PUSH VERSUS PULL

It is very natural that users treat unsolicited material with suspicion. Browsing the Internet is not generally considered a dangerous activity. In the mind of users the worst that can happen is that they could accidentally stumble on some sites of explicit nature.

The work by E.Wolak indicates that advertisements on websites are generally trusted a lot more than the same ads distributed via spamming [1]. For this very reason malware distribution via websites is more likely to be successful than by using newsgroup distribution, spamming executables or even spamming URLs to them. (*Note: for brevity, further on in this paper, we will include adware and other kinds of potentially unwanted programs into the malware category.*) For people involved in distribution of malware it makes a lot more sense to direct or entice people to their websites than to use 'push' distribution methods.

HACKING INTO WEBSITES

Let us imagine someone wants to make sure their malicious code is run by as many users as possible. They can post it on a website but, naturally, this will have very limited exposure as users are not very likely to visit a random website. This is the same problem, really, as legitimate businesses are facing – how to make sure potential customers visit their website. The main difference is that the bad guys are a lot less limited by ethical and legal boundaries.

There are several ways in which users can be diverted to a website of the attacker's choice. One is to modify a popular website to include malicious links, redirects or pop-up and pop-down windows. Frequently this attack is called 'Web defacement', even though it does not necessarily involve a modification of how the website looks (so a 'defacement' can be alien code implanted into a website and not visible by a user in a browser). It can also be an injected alien link, visible or invisible (we shall explain why links are important later). Defacement is only possible if an attacker has access (local or remote) to a website or could hack into it. Popular websites are generally more carefully maintained and their integrity is checked more frequently so attacks are less likely to succeed. However, there are still existing records of such attacks [2] and [3].

Firstly, 'defacement' attacks could be using existing vulnerabilities (so-called 'remote-root' vulnerabilities). Secondly, websites could be lacking recent security patches and prone to hacking through vulnerabilities. Thirdly, bad management and/or practices can be responsible – open shares, weak passwords, guest accounts, vulnerabilities in applications (like *Internet Explorer* if these were run by website administrators), etc.

Effects similar to the manipulation of websites can be achieved if a web proxy is hacked into. The users will see modified content even though the original website is perfectly OK. Obviously, a local malicious proxy or LSP filter (layered service provider) could have the same effect. Even though some adware is known to have done this, it is beyond the scope of this paper as malicious modifications are done locally and not to the Internet. This attack method is not yet common because the number of users served from a single proxy is usually not high. In future, however, it may grow as attempts to introduce proxy service on the Internet level are under way – for example, infamous beta of Google Web Accelerator [4]. There are additional risks in compromising websites that carry out password-caching (sites allowing users to access several bank accounts from one page or several mail accounts).

It has to be noted that subtle modifications made to a hacked website may go unnoticed for a very long time. To be able to notice a malicious change the webmaster has to perform integrity checking of the site's contents or do a manual inspection. Very few administrators do that. For big websites this is a huge task. Another method would be inspecting the logs but this is probably not the best way to find unauthorized modifications because they could have been edited out or cleared after a break-in. On the client side (a PC that contracted something from a web page) it may be difficult to trace a problem back to the source because in any average web session users frequently follow many links and visit many websites. Some defacement examples and advice on how to prevent defacements are given in [5].

We also have to mention the W32/CodeRed worms [6]. The first version of this very successful worm performed a visible defacement of a website, but a later variant [7] silently installed a backdoor program on a server (and avoided the visibility of W32/CodeRed.a). After a backdoor is installed a website is under the control of the attacker who can modify its web contents at will. The CodeRed story confirms that any zero-day web server exploit potentially provides an attacker with many thousands of web servers that could be manipulated (in case of CodeRed it was ~70,000 computers [8]). Even for known exploits, the speed of patch deployment gives attackers a window of opportunity to achieve some malware distribution before patches are universally applied.

An interesting case of using compromised computers to hide a web server (porn-related) was observed in 2003 [9]. This reverse proxy trojan was deployed on many computers and then used to route web requests to a pornographic website, thus hiding the IP address of the originating web server. This system still had a single point of failure as there was only one hidden server. Development of this idea has a lot of potential because with an army of compromised PCs one can run a distributed web server where parts of a website are split between different PCs. It would be extremely difficult to shut such a network down.

Several viruses infect new targets by mass-mailing a link to a web page that the virus has just created on a compromised computer (e.g. W32/Mydoom.ah [10]). For the W32/Mydoom.ah virus it was a simplistic http server created for only one purpose – to run an exploit and infect another machine. But it would not be very difficult to expand this concept and make this web page real. The question is then how to make sure users visit it.

In any case, adding an alien modification to legitimate sites can have only a temporary effect. If bad guys want to sustain their business they need to tap into the source. One of the best sources would be Internet search engines.

INDEX HIJACKING

The objective of this attack is to make sure a website hosting malware comes high up in the list of sites returned by an Internet search engine. That will ensure a steady supply of victims to the bad guys.

We first learnt about this attack from a user who complained that *Google* sent him to a malicious website. *Google* is very popular so we concentrated on this search engine and investigated how they rank web pages. *Google* uses a so-called 'PageRank' value to determine the quality of any web page. They state that page rank (PR) is not the only criterion and a lot of other parameters are also used. *Google* is deliberately obscure about the details: 'Due to the nature of our business and our interest in protecting the integrity of our search results, this is the only information we make available to the public about our ranking system' [11]. It is clear, however, that apart from page rank other important components include: page contents, text of the links, text around the link, contents of neighbouring pages, page URL, its filename and title. *Google* has changed its ranking strategy several times – that resulted in significant movement in the returned results as reported by Internet Search Engine Database [12].

The PR values are determined by analysing the graph representing the topology of all web pages collected by the *Google* crawler [13]. Even though this is a horrendously complex computational task, crawling the web takes even more time. On average, *Google* manages to update its ranking rules approximately once per month. Figure 1 demonstrates the Page Rank calculation method – each 'incoming' link is a 'vote' for this page that increases its PR. Each outgoing link is a vote for another page. Note: PRs are attributes of pages, not websites.

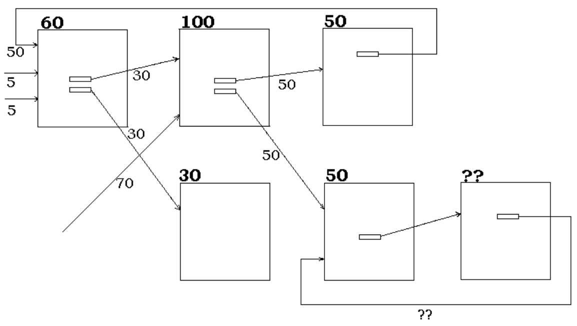


Figure 1: Page rank calculation. Numbers near pages are PageRanks (PR), numbers near links are 'PR vote' value. PR is a sum of 'PR votes'. The two pages in the bottom right corner represent a 'Rank Sink'.

There is a vulnerability in the simplistic PR approach called a 'Rank Sink'. It occurs when the graph has a loop with no outgoing links. Google has a method of handling this problem but it still can be exploited to inflate PR values by creating loops that have very few outgoing links. It can be proved that by adding good incoming links and reducing the number of visible outgoing links, one can up a PR value of a page. This is trivial to do – adding links to selected pages is easy, hiding outgoing links can be done, for example, with obfuscated scripts (instead of normal 'href' links). There are commercial companies that specialize in manipulating *Google* search results – *SubmitExpress*, *WebGuerilla* (known as SEO, or 'search engine optimization' companies). The mere existence of these companies confirms that exploitation of the ranking is possible.

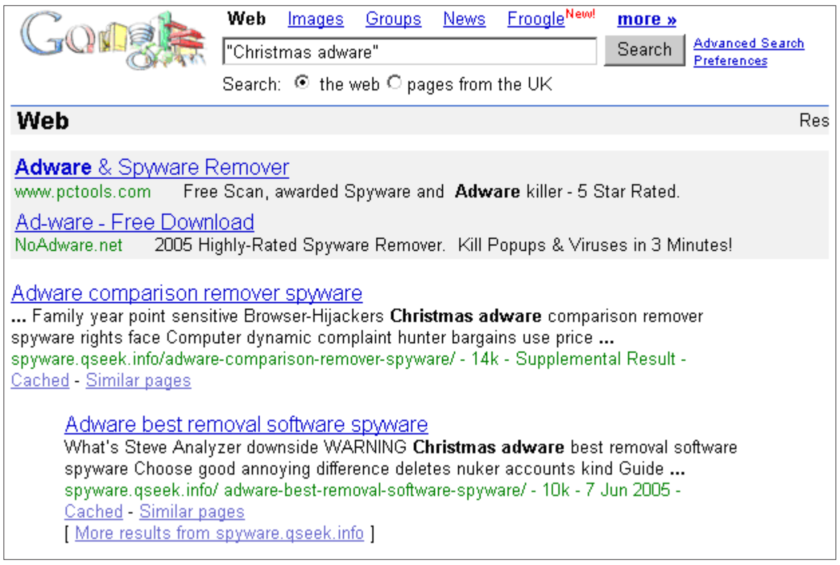
So, how do malicious attacks on *Google* trigger? For example, if a user enters into *Google* a phrase like one of these: 'Santa Trojan', 'Filmaker trojan', 'Stinger trojan', 'Skipping Christmas', 'Honda Vespa', 'crack CSS', 'Windows XP activation', 'adware Adaware', 'hacker tricks', or 'edonkey serverlist', then (s)he would get a bunch of very suspicious links. (*Important note: these are all real examples so be careful if you try. Google removed some malicious URLs from their search results but new malware-related phrases and URLs appear all the time!*) Following most of these links would load a computer with malware.

Let us follow one link. I had to find one because all those I already knew about were suppressed by *Google* after we reported them. But it was not difficult to get a hit! For example, a search for 'Christmas adware' returns a link (right after the sponsored links, at the top – see Figure 2) to 'http://spyware.qseek.info/adware-comparison-remover-spyware/'.

The contents of this web page are rather amusing and are shown in Figure 3 below. The page starts with an obfuscated redirect (remember what was said above about hiding outgoing links to create 'Page Sink' loops!) which is followed by machine-generated text (nonsense, but on the topic!). Then there is a series of links. The whole 'index.html' is ~11kb HTML, so only a small portion is presented (plus some routine HTML formatting is removed for brevity).

The text on this website is clearly machine-generated, but in such a way so that any brief computer analysis will not be able to detect that (there is proper HTML formatting, JPEG picture inclusion, links, etc.). I would be surprised if this HTML was not generated by a program that pulled most of the words from a *Google* search results for a word 'adware'! Note that the name of the link includes the keyword ('adware-comparison-remover-spyware') which makes *Google* feel it is a very relevant hit.

The phrases that trigger *Google* have to be less common so as not to drown in the useful links. On the other hand, phrases should not be unique – otherwise no user would ever look for them. The texts that are randomly assembled from words related to the topic of the page ('adware' in our case) should do well.



"Christmas adware"

Search: [✓ the web] pages from the UK

Web Res

Adware & Spyware Remover
www.pctools.com Free Scan, awarded Spyware and **Adware** killer - 5 Star Rated.

Ad-ware - Free Download
NoAdware.net 2005 Highly-Rated Spyware Remover. Kill Popups & Viruses in 3 Minutes!

Adware comparison remover spyware
... Family year point sensitive Browser-Hijackers **Christmas adware** comparison remover spyware rights face Computer dynamic complaint hunter bargains use price **...**
spyware.qseek.info/adware-comparison-remover-spyware/ - 14k - Supplemental Result -
Cached - Similar pages

Adware best removal software spyware
What's Steve Analyzer downside WARNING **Christmas adware** best removal software spyware Choose good annoying difference deletes nuker accounts kind Guide **...**
spyware.qseek.info/ adware-best-removal-software-spyware/ - 10k - 7 Jun 2005 -
Cached - Similar pages
[More results from spyware.qseek.info]

Figure 2: Google's results for 'Christmas adware' search.

```
<SCRIPT LANGUAGE=”JavaScript” ><!—//
nalco='h' + 'tt' + 'p://' + 'qs' + 'eek.info/spyware.html'; //adware-comparison-remover-spywareindex.html';
document.write('<a href=”'+nalco+'” id=”likn” target=”_self” style=display:none>go</a>');window.open(“”,
“_self”);
document.getElementById(“likn”).click();
//—></SCRIPT>

    <td><h1>Adware comparison remover spywareindex</h1>
    <p>Ad-Watch monitor feed Extensions decide DoubleClick deletes increased brand-new auto partner frequently
instead disabled ref Trade slip miss slogan. Capabilities is deletion top communication gathers Interface
prevention not Not ClickTillUWin Mozilla Allows. Time wishing However neither hosts board adware comparison
remover spywareindex offline modules Computing features Alternate Scumware Lockergnome more transferred try
hijackware Computing monthly consider beta linkdomain another Most. Blazing Adware Networks Misuse Use CSI
Updates hopeful temporary friends clean its worm User resource flavors running Press.</p>
    <h2>Adware comparison remover spywareindex two More</h2>
    <p>See describes happen checker Cleaning former plain afraid hijackers With SUGAR building qualify. Release
continuously valuable concept Imesh Spybot efforts transferred agreed Businesses each created add Cydoor Spam
well-known archive publishers strongly Nowadays. </p>
    <p align=”right”><img src=”images/adware-comparison-remover-spywareindex.jpg” alt=”adware comparison
remover spywareindex”></p>
    <h2>MAKES PRECEDED serial adware comparison remover spywareindex</h2>
    <p>Follow pop-ups content under mac acquired most BonziBuddy incarnation unrelated agendas register locat-
ing practices called auto User accurate MSIE this hopeful tremendous screens.
    <p>Started adaware developers Real OptOut Oct participate terms carried Learn Computing monitor congressman
background online haven designed time proposes helper identifiable AND.</p>

    <p>Web links: <a href=”http://spyware-removal.net.ru/”>Spyware</a>, <a href=”http://pet-supply.org.ru/”>Pet
Supplies</a>, <a href=”http://wedding-rings.qseek.info/”>Wedding Rings</a>, <a href=”http://adipex-diet-
pills.net.ru/”>Adipex</a>, <a href=”http://health-insurance-quote.fromru.com/”>Health Insurance</a>, <a
href=”http://renova.pills-center.com”>Renova</a>, <a href=”http://hydrocodone.mail333.com/”>Hydrocodone</a>, <a
href=”http://engagement-rings.qseek.info/”>Engagement Rings</a>.</p></td>
```

Figure 3: Contents of 'http://spyware.qseek.info/adware-comparison-remover-spyware/'.

This web page also changes frequently. *Google* crawler saw 'Christmas Adware' in it, but when I later checked the live page this phrase was no longer present (of course, *Google*'s cache could still show the previous version). The reason for this volatility is probably that all the pages are rebuilt as soon as page generation rules are improved. It is also an interesting observation that most similar web pages are not cached by *Google* (this can be controlled via ROBOTS.TXT file that can pass some instructions to Web crawlers).

From the results returned by 'samspade.org' it is clear that domains 'qseek.info', 'spyware-removal.net.ru', 'petsupply.org.ru', 'adipex-diet-pills.net.ru', 'pills-center.com' are all registered by the same person in Russia. You can also see a link that loops back to 'engagement.rings.qseek.info' (remember what was said about inflating PR values by creating 'Rank Sink' loops!). There are several visible outgoing links on this page and according to 'SamSpade' they all go to web pages controlled by the same people. We saw the result of these machinations – the page does come at the top of *Google* search. That confirms our hypothesis about the exploitation of 'Page Sink' loop vulnerability in *Google* PR calculations.

For other search phrases I observed similar results – *Google* ranked bad pages very high and the links went into clusters of inter-related web domains registered by the same person. One example of that was a phrase 'Filmaker trojan' that pointed to domains 'granvillas.com', 'gadalka.org', 'glastonburycc.com', 'go2resort.com', 'sunidoc.com' and 'full-circle-farm.com' all registered by Alex Kurc from Seattle. This picture is consistent with our theory of exploiting 'Sink Loops'.

LINKS HIJACKING

Let us take a look at another malicious site promoted through 'index hijacking'. I found references to this bad website from completely legitimate sites! How could this happen? We can already answer the question as to why this happened – incoming links are good for PR, so bad guys are motivated to have more of them.

A trigger was the 'Stinger Trojan' phrase in *Google*. One of the top links returned by Google was 'www.arclab.ru/stinger-trojan-removal.html'. The HTML at this URL starts with an obfuscated redirecting script:

```
<SCRIPT language=JavaScript src='inc/ojldlb.js'></
SCRIPT>
<SCRIPT language=JavaScript>
uzysaq('http://doredirect.com/
index.php?kw=spyware&id=11')
</SCRIPT>
```

The script uses an external OJLDLB.JS file, which is just another level of obfuscation – a split 'location.replace' string (clearly the *Google* crawler would not be able to recognize and follow this link!):

```
function uzysaq(imsa){
rqc="l"; uve5="oc"; mg18=1; pnw6="atio";
ssq14="n.re"; kbq18="pla";
nx26=3; wqk23="ce"; rbd26="('"; qs5=imsa; uq13="')";
if(mg18+nx26==4)
eval(rqc+uve5+pnw6+ssq14+kbq18+wqk23+rbd26+qs5+uq13);}
```

This script redirects you to 'doredirect.com' which, in turn, redirects to 'tolemon.com'.

There is an even quicker way to get to this destination. No matter what URL you try to access on 'arclab.ru' – the redirections will occur anyway! Here, for example, is a transcript of negotiations between the HTTP client (lines starting with 'C:') and server (lines with 'S:') when we try to access a non-existing HELLO.HTM. Only instead of HTML redirects (like we saw above) we will be driven by server redirects (via HTTP 'Location' function):

```
C:\>geturl www.arclab.ru/hello.htm
```

Connecting to http://www.arclab.ru:80...

```
C: GET /hello.htm HTTP/1.0
C: Host: www.arclab.ru
C:
S: HTTP/1.1 302 Found
S: Date: Fri, 27 May 2005 16:30:27 GMT
S: Server: Apache/1.3.31 (Unix)
S: Location: http://doredirect.com/
index.php?kw=spyware
S: Connection: close
S: Content-Type: text/html; charset=iso-8859-1
Connecting to http://doredirect.com:80...
C: GET /index.php?kw=spyware HTTP/1.0
C: Host: doredirect.com
C:
S: HTTP/1.1 302 Found
S: Date: Fri, 27 May 2005 16:30:27 GMT
S: Server: Apache/1.3.31 (Unix)
S: Location: http://tolemon.com/
search.php?qq=spyware
S: Connection: close
S: Content-Type: text/html
Connecting to http://tolemon.com:80...
C: GET /search.php?qq=spyware HTTP/1.0
C: Host: tolemon.com
C:
S: HTTP/1.1 200 OK
S: Date: Fri, 27 May 2005 16:37:15 GMT
S: Server: Apache/1.3.33 (Unix) PHP/4.3.10
S: X-Powered-By: PHP/4.3.10
S: Set-Cookie:
PHPSESSID=9c9d678f438496936790f174e10c6e3b; path=/
S: Expires: Thu, 19 Nov 1981 08:52:00 GMT
S: Cache-Control: no-store, no-cache, must-revali-
date, post-check=0, pre-check=0
S: Pragma: no-cache
S: Connection: close
S: Content-Type: text/html
Getting search.php (???? bytes)...
16234 bytes in 0 seconds
```

The result is the same – SEARCH.PHP page that advertises a bunch of anti-spyware programs. It presents the user with links to the following websites:

http://get.privacycash.com

http://www.STOPzilla.com

http://www.regfreeze.net

http://microantivirus.com

http://www.adultfriendfinder.com

http://alertspy.com/

www.dealtime.com

www.SpySpotter.com

If any of these links is clicked the information about who organized this click is also transmitted in the 'id=' field:

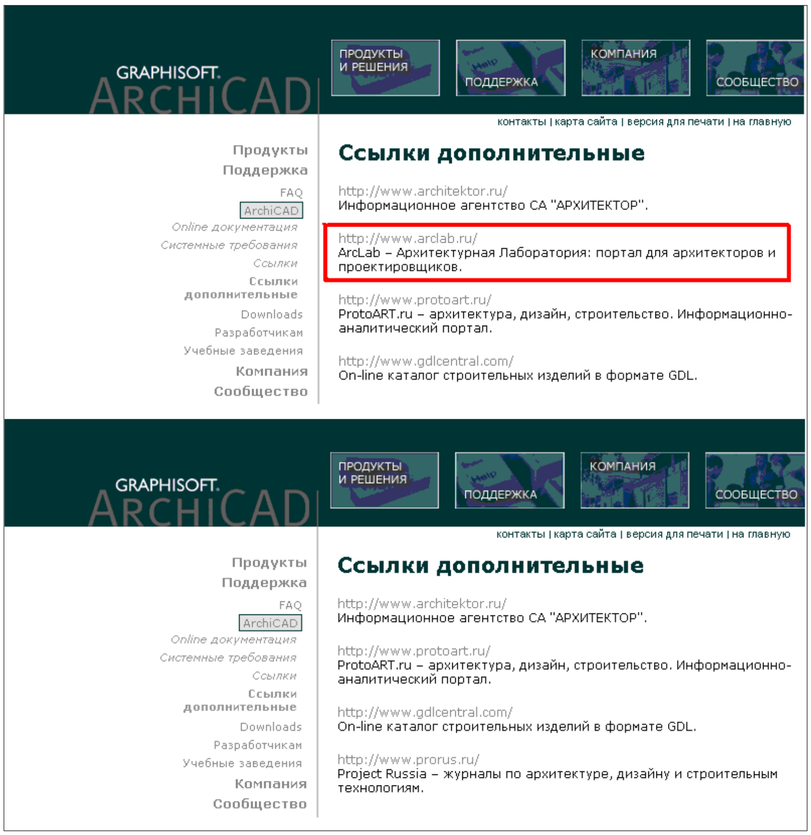
```
<a href="click.php?id=cda703d4a38549bb52d9f21f23fe92be"
<a href="click.php?id=b428d4748f0ccd5e0298cb7c25fdc9bc"
<a href="click.php?id=ab033511dcee4966449d0f56caa86ca9"
```

```
<a href="click.php?id=9b69e7811648d0d1b16869935f6df9ea"
<a href="click.php?id=d86373683ce115fc1432c87b456700b5"
<a href="click.php?id=7a418cf5f371697065b4f014bd05a83b"
<a href="click.php?id=77f7fde74137cbe5e45dc9454fa413d5"
<a href="click.php?id=b419e3790e4147db994056b6b6e235d2"
```

The purpose of this 'id=' is almost certainly pay-per-click revenue. You noticed, of course, that original search on *Google* ('Stinger Trojan') was related to the removal of malware ('Stinger' is a popular tool to remove most common malware [14]). And the user is offered what he was interested in. But not just that!

Clicking on most of those links would generate pop-under windows that point to 'spamfilter.no-ip.info' and 'buy-traffic.net/tds3/index.php'. The following exploits are used: Exploit-MhtRedir and Exploit-ByteVerify. If they are successful other malware is installed and executed (the list includes: Adware-180Solutions, Adware-WinAd, Adware-DFC, Adware-RBlast, Adware-ISTbar.b, Adware-SideFind, Adware-ValueAd, Adware-TopRebates, Adware-Qoolaid and Trojans VBS/Inor, Downloader-JU and Downloader-UI).

It is curious that the domain 'doredirect.com' is registered by 'Vasiliy Pupkin from Muhosransk' ('tolemon.com' is registered by 'Muhosransk from Muhosransk'). Both names have a lot to say to any native Russian speaker – they sound just like, for example, 'Mickey Mouse from Dogspoo city' would sound to a native English speaker. It could be a coincidence but a string 'vvpupkin' is found in HTML samples related to BackDoor-AXJ (aka Berbew). This is exactly the same backdoor that was involved in the case of web servers with IIS vulnerability [2].



Ссылки дополнительные

http://www.architektor.ru/
Информационное агентство СА "АРХИТЕКТОР".

http://www.arclab.ru/
ArcLab – Архитектурная Лаборатория: портал для архитекторов и проектировщиков.

http://www.protoart.ru/
ProtoART.ru – архитектура, дизайн, строительство. Информационно-аналитический портал.

http://www.gdlcentral.com/
On-line каталог строительных изделий в формате GDL.

Ссылки дополнительные

http://www.architektor.ru/
Информационное агентство СА "АРХИТЕКТОР".

http://www.protoart.ru/
ProtoART.ru – архитектура, дизайн, строительство. Информационно-аналитический портал.

http://www.gdlcentral.com/
On-line каталог строительных изделий в формате GDL.

http://www.prorus.ru/
Project Russia – журналы по архитектуре, дизайну и строительным технологиям.

Figure 4: A web page from 'www.archiCAD.ru' before and after removing the link to 'arclab.ru'.

Now, the most interesting part – I found links to 'arclab.ru' on a completely legitimate Russian Internet portal for architects and designers 'www.archiCAD.ru' (it seems like a big and popular site!). On a page of useful external links 'arclab.ru' is described as a place for discussions about architectural projects. Nevertheless, 'arclab.ru' is a malicious site. I contacted the domain administrator for 'archiCAD.ru' and he reported that 'this bad link was removed' (see Figure 4). One possibility is that the 'www.archiCAD.ru' server was hacked into and alien links were injected. Another possibility is that the domain 'arclab.ru' expired and was registered by the bad guys (in a word – hijacked). I tend to believe it was indeed hijacked because there seem to be so many references to 'arclab.ru' from other sites. We now know precisely a reason why someone would like to hijack a popular domain – any such site will still have a lot of incoming links and they are a commodity from the point of view of web ranking!

People behind 'index hijacking' would very much like to point a link from a popular website into their own 'Page Sink' exploit loop. That would boost PR 'votes' and fool *Google*. Even links coming from less popular sites would help a lot with the ranking if there were really many of them. It does not matter how such links are introduced (hacking, backdoor or domain hijacking) – it is the links themselves that are useful.

So, there is likely a new, previously untapped, reason for deploying backdoors – to steal the ranking of web pages by injecting links after gaining access to web servers!

DNS POISONING (PHARMING)

As you know DNS servers are responsible for the translation of symbolic names (like 'www.ibm.com' to numeric IP addresses like '129.42.16.99'). Access to almost any resource in the Internet requires such a conversion. If an incorrect conversion takes place a user will end up accessing a different resource. There is a certain similarity to companion viruses in a file system where original filename points to a virus instead of the original file. (The role of DNS is taken here by the file system that translates a symbolic filename into a numeric disk cluster number). Clearly, this is a gold mine for distributing malware!

There are two rather different kinds of DNS poisoning. The first one is when authoritative DNS data (stored on the DNS server's hard disk) is modified. The second is when only a temporary DNS cache data in memory is poisoned.

The first scenario is significantly worse because the table modification may be replicated to other DNS servers. There is a hierarchy of DNS servers (it is related to the hierarchy of zones for which they are responsible) and any modification of DNS tables on a higher level of this hierarchy will be propagated (usually within ~24 hours) to many other DNS servers that are on a lower level. If an attacker succeeds in modifying DNS tables he can direct all the users of poisoned DNS servers to any IP address of their choice.

There are several ways to introduce a malicious DNS modification – exploits in DNS protocol, hacking into a DNS server, social engineering.

Exploits in the protocol allow an attacker to read, intercept and modify DNS information when it is passed between DNS servers.

DNS poisoning is not a new phenomenon. Weaknesses in BIND (which is a Unix-based tool providing DNS functionality for the majority of Internet DNS servers, standing for 'Berkeley Internet Name Domain') have been discussed in public for over 15 years – in 1989 [15], 1993 [16] and 1997 [17]. The weakness described by Schuba is related to poisoning BIND's DNS cache (all DNS implementations use caching to achieve better performance and can return DNS data based on cache rather than authoritative data that is not in the cache). The CERT advisory described a weakness in BIND related to the fact that DNS transaction ID numbers were sequential. Because they were sequential an attacker could pick the next ID and spoof a transmission from a trusted DNS server. Such an attack would work particularly well if an attacker can sniff the traffic of the DNS server under attack. The easy solution to the problem of predicting transaction IDs was to randomize them. This was released as a patch to BIND. Later, weaknesses were discovered in the randomization routines that still allow an attacker to predict the next ID. It is also known that brute-force attack (aka 'Birthday attack') with random IDs has a fairly high chance of success [18, 19]. To mitigate attacks based on sniffing and spoofing of the DNS messages authentication and encryption have to be built into the DNS protocols (that is 'DNSSEC' initiative [20]).

Attacks on DNS servers can be based on the 'Ask Me' approach [16]. The idea is to get the victim DNS server to send a DNS query to the DNS server under control of the attacker. Then the reply from the attacker's DNS server can include poisoned information which will stay in the DNS cache of the victim DNS server for at least a while. To trigger such a query the attacker can, for example, send an email to a wrong email address within the zone of the victim DNS server. That will generate a DNS query from the victim server to the attacker's DNS server because the victim server will need to get the DNS information to send the non-delivery mail message.

Hacking into a DNS server potentially gives an attacker full control over DNS tables. Such a hacking can, for example, be done remotely through a successfully deployed rootkit or backdoor. A brute-force login attack is another possibility (an army of bots may be able to do that well enough). Spoofing a legitimate domain owner via a phone, fax or email could work too [21]. This sort of attack is sometimes called domain hijacking.

Apart from BIND there is also *Microsoft*'s implementation of DNS for servers running *Microsoft* OSs. Obviously, weaknesses in the DNS messaging protocol apply equally to Unix and *Microsoft* versions. There was, however, an additional DNS caching problem for *Windows NT 4.0* and *Windows 2000* [22]. Remedies are described in [23]. Some gateway products, firewalls and appliances are also susceptible to DNS poisoning attacks [24]. Fixes are available from the manufacturer [25].

One also has to be aware of the fact that a lot of contemporary malware and adware modify local HOSTS/RHOSTS files, resulting essentially in a local DNS poisoning. That means the Internet DNS system may work perfectly OK but it would never get a chance because the incorrect DNS resolution occurs locally. It has not been yet observed in the field but it is perfectly possible for malware to intercept read requests to the HOSTS file and poison the data. Physical modification of the HOSTS file (with stealthing the modifications) is, of course, even better as it will be effective even in safe mode or when the malware is removed (until the HOSTS file is cleaned). A lot of malware (and IRC bots in particular) have a habit of modifying the HOSTS file to redirect IP addresses associated with AV/security sites and stop security programs from updating themselves.

After DNS poisoning is discovered it could take significant time and effort to fix the problem. That is due to the distributed nature of the DNS system and significant delays in refreshing DNS tables because the changes have to propagate through the entire network of DNS servers. But it is not easy to discover the problem in the first place because poisoning may appear as a non-reproducible problem due to refreshing of the cache and due to expiration of the poisonous records (when its TTL = 'time to live' expires). That means an inspection of a DNS server can reveal correct behaviour but the next minute the same server may be poisoned again. DNS software, obviously, need to be updated to the latest version that includes the patch.

In Jan–May 2005 there were several large-scale DNS poisoning attacks. One of them resulted in the redirection of at least 1,304 popular domains [26]. Installation of malware was achieved automatically (by just browsing to a website) through several known *Internet Explorer* vulnerabilities. The following malware and adware was involved:

Exploit-MhtRedir.gen	Adware-Websearch
Exploit-ANIfile	Adware-SAHAgent
AdClicker-CN	Adware-WinAd
AdClicker-AF.dr	Adware-DFC
AdClicker-AF	Adware-RBlast
Downloader-TD	Adware-ISTbar.b
Downloader-YN.dr	Uploader-R
Adware-180Solutions	Uploader-R.dr
Adware-SideFind	PowerScan
Adware-Websearch.dldr	

Detailed analysis done by [27] shows that DNS poisoning is frequently driven by the money received by the bad guys from advertising companies on a pay-per-click basis.

Very recently, the media started using the term 'pharming' to describe DNS poisoning [28]. That was obviously inspired by 'phishing' attacks although the two techniques have very little in common. There is a nasty possibility, though, of using DNS poisoning for phishing. If DNS records for popular banks are poisoned, even if a user goes to a correct banking site s(he) could be redirected to malicious websites masquerading as real ones. There is very little that can be done to counter such an attack (short of hard-coding IP addresses – not very user-friendly!). The problem is that authentication of whether the target website is genuine is fairly weak. Manual inspection of the site's security certificate (https) would work but many users are likely to miss even the fact that a site is not using encrypted (https) communication! Yes, users frequently make mistakes…

EXPLOITING USERS' MISTAKES

How frequently, on average, do we mistype a URL? I think I would be conservative in an estimate that 1 in 10 URLs is typed incorrectly. With clever companies no matter how you twist the name you will find the right site (kelkoo.com = kelcoo.com, kellkoo.com, kellku.com, kelku.com, kelcoo.com, celcoo.com; plus '.co.uk' and '.org' would also work). But when millions of users access millions of sites mistakes will occur frequently and people will come to sites they never intended to visit. Most of such mistakes are, of course, pretty easy to correct. Many mistyped URLs bring us only to placeholder web pages offering to resell a domain (check 'www.simantek.com' or have a look at a funny example – 'www.mikrosoft.com').

It is fun to try to access URLs: 'www.ibm.com', 'www.iibm.com', 'www.ibbm.com', 'www.ibmm.com' and see what happens! Would you expect to be offered to change your browser's HomePage to 'www.munky.com' picturing a playful ape?

Capturing misspelled domain names is very common and there is even a name for this activity – 'typosquatting' (a derivative of 'cybersquatting'). All these domains: 'www.macafee.com', 'www.mcafeee.com', 'www.mcafe.com', 'www.mkafee.com' (which, incidentally, is the same as 'www.mycrosoft.com'!), 'www.symantek.com', were clearly registered to capture the attention of people who accidentally made a typo. Some of the people may not immediately realise that they had a finger trouble because all of these sites are actually related to security software.

But if we type the domain 'www.semantec.com' (or 'www.simantec.com') we would suddenly enter a grey zone – a numeric IP name that is clearly related to advertising. If you click 'www.makafee.com' it will redirect you to 'www.ownbox.com/treasure/McAfee.html' (ownbox – may seem a bit scary!). This, in turn, will redirect to 'http://click.linksynergy.com/fs-bin/click?id=HJoMP0cKO0s'. From there you will actually get to the 'http://us.mcafee.com' page (after all, this turned out to be just a benign pay-per-click scheme).

There are three basic sources of 'typosquatting' – common misspellings (e.g. 'webadress'), typos (e.g. 'wwebaddress') and different representation (e.g. 'web-address'). There is even software for ISPs from a company 'Paxfire' that redirects most common mistyped domains for advertising purposes. There is also a shareware tool called 'Snapfiles' to locate the most commonly misspelled domain names and you can register your favourite choice online.

One would expect that typosquatting activity can also be used to distribute malware and adware. And that is true.

Site	**Content**
'www.whitehouse.org'	satirical
'www.whitehouse.com'	porn
'www.wilipedia.org'	adware
'www.wiipedia.org'	adware
'www.eikipedia.org'	adware
www.googkle.com'	adware+malware

The most high-profile case of such misuse was recorded at the end of April 2005 when 'www.googkle.com' was installing a host of malware and adware utilizing a series of different exploits. The list of installed malware is impressive:

Exploit-MhtRedir.gen	Downloader-UV
VBS/Psyme	Generic BackDoor.u
Downloader-GS	BackDoor-AWV
Spabot	Downloader-UV
Spabot.dll	BackDoor-AML
Downloader-XB	Adware-NSearch
StartPage-GT	Adware-IEToolBar.dr
PWS-Banker	Adware-NSearch
Proxy-TSOH.dll	

Even one fully-functional backdoor is, actually, enough as the PC will be in the hands of the attackers.

ACKNOWLEDGEMENTS

I am grateful to Matt Cutts of *Google Inc.* who provided some information for this research. I would also like to thank my colleagues at *McAfee*: Federico Barbieri, Daniel Wolff and Fraser Howard for fruitful discussions and Dmitry Gryaznov for his excellent URL analysis tool. I also appreciate input from Lee Fisher.

CONCLUSIONS

Spammers used malware to create armies of zombie proxies (with W32/Sober and W32/Bagle, for example) to carry on with spamming (and that is with spamming being illegal in many countries!). Adware distribution is not illegal and the annual value of online advertising business is exceeding one billion dollars. It is not surprising that a lot of malware is involved in the distribution of these dubious programs. Phising is also very profitable. All in all, malware distribution has solid monetary support and the stakes in adware and phishing are a lot higher than in the spam game.

A few years ago everybody just loved email. Now, due to severe spam attacks, this communication channel is rapidly losing its popularity. Email is suffering from an ever-increasing noise/signal ratio (more spam in your inbox) and dropping reliability (more and more frequently important mails are lost due to false positives in anti-spam programs). Many people are switching to white-listing email senders.

I predict that the same fate is awaiting the web because the attacks on the Internet have already started and the ferocity of them is rapidly increasing. We expect this to become a more common malware distribution vector. We are likely to see browsers that operate whitelists of websites soon. What used to be a playfield for a bunch of schoolboys and script kiddies is now an arena for gangs that have serious commercial interests at stake.

Is it not ironic that when I was looking for evidence of 'index hijacking' and searched for 'Google+SEO' (SEO = search engine optimization) in *Google* one of the returned links was a manipulated one ('http://resignation-sample.strettyp.com/') and attempted to install Adware-WinAd on my machine using an exploit. That is what I would call 'index hijacking' in action!

REFERENCES

[1] Wolak, Chaelynne; 'Advertising on the Internet', http://www.itstudyguide.com/papers/cwDISS890A3.pdf.

[2] http://vil.nai.com/vil/content/v_100488.htm, http://www.lurhq.com/berbew.html, http://www.microsoft.com/security/incident/download_ject.mspx.

[3] http://news.yahoo.com/s/ap/20050603/ap_on_hi_te/microsoft_hacked.

[4] http://www.windowsdevcenter.com/pub/a/windows/2005/05/24/google_accelerator.html.

[5] http://www.sans.org/CDIeast/deface.pdf.

[6] http://vil.nai.com/vil/content/v_99142.htm.

[7] http://vil.nai.com/vil/content/v_99142.htm.

[8] Gryaznov, Dmitry; 'Red Number Day', *Virus Bulletin*, October 2001, p.8. http://www.nai.com/common/media/vil/pdf/dgryaznov_VB_oct2001.pdf.

[9] http://vil.nai.com/vil/content/v_100480.htm.

[10] http://vil.nai.com/vil/content/v_129631.htm.

[11] http://www.google.com/corporate/tech.html.

[12] http://www.isedb.com/news/article/663.

[13] Page, Larry; Brin, Sergey; Motwani, R.; Winograd, T.; 'The PageRank Citation Ranking: Bringing Order to the Web', http://citeseer.ist.psu.edu/page98pagerank.html.

[14] http://netforbeginners.about.com/od/readerpicks/l/bltop10freeware.htm.

[15] Bellovin, Steven; 'Using the Domain Name System for System Break-Ins", *Proceedings of the Fifth Usenix Unix, Security Symposium*, June 1995.

[16] Schuba, Christoph; 'Addressing Weaknesses in the Domain Name System Protocol', http://ftp.cerias.purdue.edu/pub/papers/christoph-schuba/schuba-DNS-msthesis.pdf.

[17] Advisory CA-1997-22, 'BIND – the Berkeley Internet Name Daemon', http://www.cert.org/advisories/CA-1997-22.html.

[18] Vulnerability Note VU#457875, URL: http://www.kb.cert.org/vuls/id/457875.

[19] Stuart, Joe; 'DNS cache poisoning', http://www.securityfocus.com/guest/17905.

[20] http://www.dnssec.net/.

[21] Sax, Doug; 'DNS Spoofing (Malicious DNS Poisoning', http://www.giac.org/certified_professionals/practicals/gsec/0189.php.

[22] http://support.microsoft.com/kb/316786/EN-US/.

[23] http://support.microsoft.com/default.aspx?scid=kb;en-us;241352.

[24] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-0817.

[25] http://securityresponse.symantec.com/avcenter/security/Content/2005.03.15.html, http://securityresponse.symantec.com/avcenter/security/Content/2004.06.21.html.

[26] http://isc.sans.org/presentations/dnspoisoning.php

[27] 'Pay-Per-Click Hijacking', http://www.lurhq.com/ppc-hijack.html.

[28] Vamosi, Robert; 'Alarm over pharming attacks', http://reviews.cnet.com/4520-3513_7-5670780-1.html?tag=nl.e497.