# Have you been Confickered?

*Martin Overton*
*Independent Researcher, UK*
*9th March 2009*

The last year or so has been rather quiet in the area of worms; not since the likes of Nimda and its offspring have we seen anything like the level of infection or disruption that a certain family of worms caused during the build up to Christmas 2008, and well on into the New Year celebrations of 2009.

In fact, as I write this article, a new rewrite with extra functionality has been found in the wild.

This article will cover what Conficker[1] is, how it works, why it managed to infect as many as 10 Million systems [*if you believe the numbers being bandied about*], and how to fight and defeat it. Finally, what lessons need to be learned to help minimise a similar outbreak in the future. But first some background:

"*On October 23, 2008, Microsoft released a critical security update, MS08-067, to resolve a vulnerability in the Server service of Windows that, at the time of release, was facing targeted, limited attack. The vulnerability could allow an anonymous attacker to successfully take full control of a vulnerable system through a network-based attack, the sort of vectors typically associated with network worms.*"

## Give me an A…

Conficker.A first showed up at the end of November 2008 and it looks like it may have been created on the 21st of November when exploit code for MS08-067 was available! So, it wasn't even a malware using zero-day exploit code.

As new malware goes, it was a relatively damp squib, and was just seen as another of the many thousand new samples seen during that period. Why? Well, it mainly relied on getting onto a system using exploit code for a flaw in SVCHOST.EXE. So, as most home users will have patched immediately[2], leaving just corporate and academia systems as potential targets. That is unless they were in the Ukraine or using a Ukrainian keyboard at which point the malware would simply exit without infecting the system.

How did it work out where the infected system was?

Simple, Conficker uses the following URLs to determine the computer's geographic location:

- *getmyip.org*
- *getmyip.co.uk*
- *checkip.dyndns.org*

If the MS08-067 vulnerability is successfully exploited on the system, Conficker.A gets the infected system to download a copy of the worm from the host computer via the HTTP protocol using the random port opened by the worm.

If the date is after November 25, 2008, but before the 1st of December 2008, Conficker builds a URL in the following format and attempts to download a file from it:

*<random ip?>/search?q=%d&aq=7*

[1] Also known as Downadup, Kido, Agent…
[2] The small number of home users infected seems to confirm this, surprising, I know.

Other than that it had no other main infection vector[3]. Once installed on a system it would install itself as a service, open a backdoor [a HTTP server] and scan the network for other vulnerable systems to infect. Oh, and to makes restoration more difficult it would attempt to reset the system restore points, potentially defeating recovery using this feature of the operating system.

If the date is on or after 1 December 2008, it connects to the domain *trafficconverter.biz* and attempt to download and execute a file found there [loadav.exe].

In fact that it how it was given the name Conficker[4]; *con* and *er* from converter, *fic* from traffic, the *k*, seems to have been an after thought, making *con-fic-k-er*.

## Next is B…

I actually saw this new variant on the 30th[5] of December 2008, at that time only 3 out of 32 tested anti-malware tools were detecting it, so it was very, very new.

So, I did some analysis of the sample and quickly found that it was *VMWare* aware, and actually changed its behaviour; instead of it infecting the system the malcode process simply went to sleep instead.

Furthermore, some of the results I had back seemed to indicate that it is also aware of sandboxes too, which is quite unusual and certainly not a common feature of most malcode.

Analysis was further complicated as the malware file is actually a DLL[6] [Dynamic Load Library] and can't be simply executed; it needs to be loaded and executed in a different manner instead.

The good news was that the sample did not appear to be polymorphic, which made some remediation and ad-hoc detection potentially easier until the rest of the anti-malware/security products were updated to detect and remove it.

## What a tangled web we weave…

However, the more I dug and the more data I was gathering about this new variant, the more I realised that this was going to be a major problem to many, many organisations and academia too. Why? Well not only was it using the MS08-067 exploit code, ala variant A, it also carried lots of other propagation/infection methods that the initial variant lacked.

So, what were these new methods and why were they going to cause so many system and network administrators so much trouble over the coming weeks and beyond?

**Patch me if you can…**

So, you may be wondering why it seems that only computers in academia and organisations are being troubled by Conficker, especially the B variant. The answer is simple, as both the A and B variants use exploit code for MS08-067, most organisations and academia tend to have long patch cycles, typically months. Whereas, most home users seem to update their systems more frequently, typically days or weeks[7]. This ensured that Conficker would be more successful in infecting non-home-user systems.

But, this isn't the only way that Conficker.B could spread, and these other infection vectors would cause far more trouble and have a longer lasting impact than anyone could have reasonably foreseen.

[3] More details on this variant can be found here:
http://www.microsoft.com/security/portal/Entry.aspx?Name=Worm:Win32/Conficker.A
[4] The name was coined by Microsoft analyst Josh Phillips.
[5] Although the sample I had was captured on the 29th.
[6] The A variant was also a DLL.
[7] Even if they don't realise it is happening automatically.

**Look-out for Lockouts…**
One of the first signs that Conficker.B had penetrated your corporate or academic network was that the help desk were being flooded by calls about system [account] lockouts[8], and no obvious reason could be found for this happening.

Certainly, it was highly unlikely that 70-90 percent of your network users had suddenly developed amnesia at the same time, and as a result typed in the wrong password or user id?

So, what was causing it?

Well, it was a particular function of Conficker.B. Once a system was infected it would scan the network for new systems to infect, first trying to exploit vulnerable system via MS08-067, and if that failed it would then try a dictionary attack to remotely login to the system so that it could infect it via the ADMIN$ share. To attempt this, Conficker.B carried a large password list[9] in its code. The dictionary attack was what was causing the account lockups on many networks.

If the dictionary attack was successful, the remote system would be infected as would any network or local shares that the system had access to.

| | | | | | |
|---|---|---|---|---|---|
| fuck | zzzzz | zzzz | zzz | xxxxx | xxxx |
| xxx | qqqqq | qqqq | qqq | aaaaa | aaaa |
| aaa | sql | file | web | foo | job |
| home | work | intranet | controller | killer | games |
| private | market | coffee | cookie | forever | freedom |
| student | account | academia | files | windows | monitor |
| unknown | anything | letitbe | letmein | domain | access |
| money | campus | explorer | exchange | customer | cluster |
| nobody | codeword | codename | changeme | desktop | security |
| secure | public | system | shadow | office | supervisor |
| superuser | share | super | secret | server | computer |
| owner | backup | database | lotus | oracle | business |
| manager | temporary | ihavenopass | nothing | nopassword | nopass |
| Internet | internet | example | sample | lovel23 | bossl23 |
| workl23 | homel23 | mypcl23 | templ23 | testl23 | qwel23 |
| abcl23 | pwl23 | rootl23 | passl23 | passl2 | passl |
| adminl23 | adminl2 | adminl | passwordl23 | passwordl2 | passwordl |
| default | foobar | foofoo | temptemp | temp | testtest |
| test | rootroot | root | adminadmin | mypassword | mypass |
| pass | Login | login | Password | password | passwd |
| zxcvbn | zxcvb | zxccxz | zxcxz | qazwsxedc | qazwsx |
| qlw2e3 | qweasdzxc | asdfgh | asdzxc | asddsa | asdsa |
| qweasd | qwerty | qweewq | qwewq | nimda | administrator |
| Admin | admin | alb2c3 | lq2w3e | 1234qwer | 1234abcd |
| l23asd | l23qwe | l23abc | l23321 | l2321 | l23l23 |
| 1234567890 | 123456789 | 12345678 | 1234567 | 123456 | 12345 |
| 1234 | 123 | | | | |
| 99999999 | 88888888 | 77777777 | 66666666 | | |
| 9999999 | 8888888 | 7777777 | 6666666 | | |
| 999999 | 888888 | 777777 | 666666 | | |
| 99999 | 88888 | 77777 | 66666 | | |
| 9999 | 8888 | 7777 | 6666 | | |
| 999 | 888 | 777 | 666 | | |
| 99 | 88 | 77 | 66 | | |
| 9 | 8 | 7 | 6 | | |
| 55555555 | 44444444 | 33333333 | 22222222 | | |
| 5555555 | 4444444 | 3333333 | 2222222 | | |
| 555555 | 444444 | 333333 | 222222 | | |
| 55555 | 44444 | 33333 | 22222 | | |
| 5555 | 4444 | 3333 | 2222 | | |
| 555 | 444 | 333 | 222 | | |
| 55 | 44 | 33 | 22 | | |
| 5 | 4 | 3 | 2 | | |
| llllllll | 00000000 | 0987654321 | | | |
| lllllll | 0000000 | 987654321 | | | |
| llllll | 00000 | 87654321 | | | |
| lllll | 0000 | 7654321 | | | |
| llll | 000 | 654321 | | | |
| lll | 00 | 54321 | | | |
| ll | | 4321 | | | |
| l | | 321 | | | |
| | | 21 | | | |
| | | 12 | | | |

*Fig 1 Conficker.B Password List*
*(Source: SOPHOS)*

[8] Assuming you had a lockout policy in place, you do, don't you?
[9] The full list contains 250 weak passwords, hopefully yours isn't in them?

**Where am I, and what's the date?**
Once Conficker.B is installed it will try to find out the public IP address of the infected computer by visiting one or more of the following websites:

*www.whatsmyipaddress.com*
*www.getmyip.org*
*www.whatismyip.org*
*checkip.dyndns.org*

Once that is done it will want to know the date for another internal routine, to do this it will access one of more of the following sites, rather than relying on the systems supposed date and time:

*www.myspace.com*
*www.ebay.com*
*www.msn.com*
*www.cnn.com*
*www.aol.com*

**Autorun if you dare…**
Many organisations and most academic campuses have poor or non-existent controls for removable media and devices. After all, these are frequently seen as the successor to the floppy disk which has almost totally disappeared as a storage method by the late 90s.

The new replacements to the venerable floppy disk include, USB thumb drives, SD/MMC and similar flash cards, USB external hard disks, Media players such as iPods and of course mobile phones and smart-phones.

The problem with many of these devices is that Windows will happily read and execute any *Autorun.inf* file found on them by default. This means that simply inserting a USB thumb drive, iPod or other supported device or storage card can automatically infect the computer. However, this is not the only risk as if the system is already infected by malcode that uses Autorun it will become infected[10] allowing the malcode to spread via sneaker-net, just like floppy disks did in the 1980s and 90s; a real case of Déjà- vu for many of us that have been working in security for two or more decades.

Autorun wasn't just used as an infection vector on removable drives and media; it was also used on Windows shares, including remote ones that the infected system had write-access to.

When a shared drive or removable media is infected Conficker.B uses the following method to drop the file:

*<Drive>:\RECYCLER\S-%n-%n-%n%n-%n%n%n%n%n%n%n%n%n-%n%n%n%n%n%n%n%n%n-%n%n%n%n%n%n%n%n%n-%n%n%n%n%n%n%n%n%n-%n%n%n%n\<random>.<ext>*

Where *%n* is a random numeric digit and *<Drive>* is the relevant drive letter. It then creates an obfuscated AUTORUN.INF file in the root of the drive to auto-run the worm whenever the drive is accessed.

It also tries to use social engineering the end users by changing the icon in the AutoPlay dialogue box, see Fig 2.

[10] Assuming it is a writable medium that has not been write-protected before insertion.


*Fig 2 Conficker.B Autorun in action*
*(Source Microsoft)*

**Kill Bill and security too…**
To protect itself and to make remediation more complex, Conficker.B resets the systems restore points [as did A], and then goes on to disable a number of key system services:

- *Windows Security Center Service (wscsvc)* which notifies users of security settings (e.g. Windows Update, firewall and anti-virus).
- *Windows Update Auto Update Service (wuauserv).*
- *Background Intelligence Transfer Service (BITS)* which is used by Windows Update to download updates using idle network bandwidth.
- *Windows Defender (WinDefend).*
- *Error Reporting Service (ersvc)* which sends error reports to Microsoft.
- *Windows Error Reporting Service (wersvc*).

If this isn't enough it also locks the memory resident copy of itself so that removal is trickier.

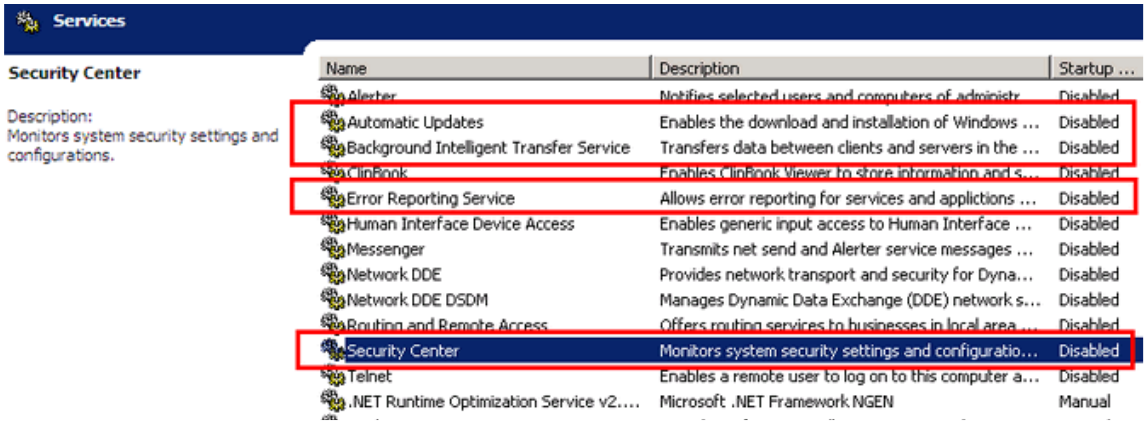


*Fig 3: Conficker.B Disabled Services*
*(Source: CA)*

To ensure that you can't easily get any help on this infection it monitors DNS requests for any URL that contains the following:

| | |
|---|---|
| *ahnlab* | *k7computing* |
| *arcabit* | *kaspersky* |
| *avast* | *malware* |
| *avg.* | *mcafee* |
| *avira* | *microsoft* |
| *avp.* | *nai.* |
| *bit9.* | *networkassociates* |
| *ca.* | *nod32* |
| *castlecops* | *norman* |
| *centralcommand* | *norton* |
| *clamav* | *panda* |
| *comodo* | *pctools* |
| *computerassociates* | *prevx* |
| *cpsecure* | *quickheal* |
| *defender* | *rising* |
| *drweb* | *rootkit* |
| *emsisoft* | *sans.* |
| *esafe* | *securecomputing* |
| *eset* | *sophos* |
| *etrust* | *spamhaus* |
| *ewido* | *spyware* |
| *f-prot* | *sunbelt* |
| *f-secure* | *symantec* |
| *fortinet* | *threatexpert* |
| *gdata* | *trendmicro* |
| *grisoft* | *vet.* |
| *hacksoft* | *virus* |
| *hauri* | *wilderssecurity* |
| *ikarus* | *windowsupdate* |
| *jotti* | |

Just like Conficker.A, the worm installs itself as a service to ensure that it automatically gets loaded when the operating system starts. However, it sets up a scheduled task to run the malcode; sort of belt and braces for infection, see Fig 4.

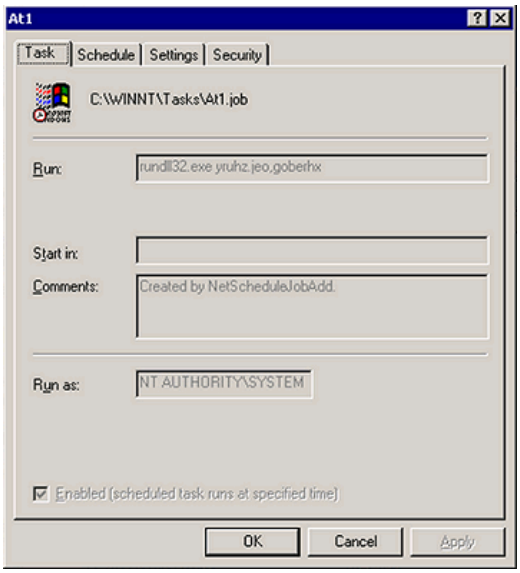


*Fig 4: Conficker Scheduled Task (Source CA)*

To make sure that Conficker isn't restrained by Windows TCP/IP stack throttling, it executes the following command on infected systems. If the operating system is Windows Vista or Windows Server 2008:

*netsh interface tcp set global autotuning=disabled*

Which disables Windows auto-tuning.

It then sets the following registry entry to allow multiple simultaneous connections on the infected system [XP or later]:

*HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpNumConnections = 0x00FFFFFE*

**Update me…**
Just to add more fuel to the fire and to help it retain control of the infected systems, Conficker.B has an interesting auto-update function. Normally this meant that the malcode would contain a download URL or similar link inside the binary to get updates. Problem is that if that link is discovered then it is relatively simple to block the download site via a black-holed DNS entry or firewall block.
Those behind Conficker.B decided on a novel method of getting round this problem. Instead of a hard coded link, Conficker.B can get updates by using an algorithm which generates a daily list of 250 download locations to use; these change everyday.

The constructed URL looks like this:

*http://<generated-host>/search?q=<counter>*

This approach allows Conficker not only to be updated, but to also be able to function like a botnet.

The downloaded file is not blindly run, it is authenticated as valid via an MD6 signature and the file is also encoded. The reason for this is to try and stop the botnet that the Conficker authors are creating from being hijacked or disabled; either by other bad guys and girls, or indeed anyone from the white hat brigade.

The latest version uses a modified algorithm for generating download sites, instead of 250 download locations a day being generated, the new version uses a new 50,000-a-day domain generation algorithm, as well as a possible 116 domain suffixes.

Why is it doing that? The simple answer is to make it as hard as possible for the good guys and girls that have been pro-actively registering as many of the 250 a day domains as they could to try and hobble or thwart the authors of Conficker. There is no way they are going to be able to fund even one percent of the costs associated with the 50,000 a day that this variant generates. In fact I don't expect the authors of Conficker to register more than a quarter to a half of a percent on these new ones. If they are smart they will only register a random selection of them each day.

This new variant also seems to be more interested in keeping hold of systems rather than infecting new ones, at this time. It is also more aggressive as it is targeting more security and anti-malware tools and if found it tries to kill it[11]. According to Symantec this new variant is not the so-called B++ one but a later and further improved version instead.

## So, what can we do?

Let me cover some generic ways that you can not only help minimise the risk of being infected by Conficker, but also other malcode that uses many of the same techniques.

**Close the Windows**
A good idea for not only minimising the risk and impact of malcode, but also unwanted or unapproved applications on your systems is to restrict the privileges that the account has for day to day use. This

[11] More details can be found here: https://forums2.symantec.com/t5/blogs/blogarticlepage/blog-id/malicious_code/article-id/249 and here: http://www.symantec.com/business/security_response/writeup.jsp?docid=2009-030614-5852-99

could be achieved via a local or domain policy, or indeed by just making the user account just that, a user account, rather than an administrator one.

**Hobble Autorun**
It isn't only Conficker that misuses the Autorun functionality of Windows, many malware now do as a matter of course. So, unless you have a genuine business need, I would very, very strongly suggest that you *fully* disable it using the *SYS:DoesNotExist* method.[12]

**Patch, Patch, Patch**
If you are in charge of a large network you should seriously look at speeding up your patch testing and roll-out methodologies and timelines. Pushing updates out months after they have been released is generally a bad idea which plays into the hands of the bad guys and girls. Don't make it easier for them.

**IPS and Firewall and IDS, oh my!**
Most malware nowadays has phone home, auto-update and the ability to spread via both local and remote network links. So, why not ensure that you can at the very least monitor what bad traffic is traversing or originating on your network via the use of IDS and strict firewalling. For the more adventurous of you that wish to be more proactive take a good long hard look at IPS and NAC solutions. IPS and NAC are active protection solutions for your networks and can, if properly implemented stop new malware in their tracks, or even from getting onto your network in the first place.

**Need I say it?**
I really hope not, but here goes; just to be sure…protect all your systems, not forgetting *NIX and Mac based systems from malware by using good up-to-date and enabled anti-malware tools and personal firewalls too.

Why mention Mac and *NIX systems, well if you run Samba on them then many share-crawling worms will happily drop infected files onto the shares, after all Samba is pretending to be a Windows file server, isn't it?

## Conclusions

It seems that many of the lessons that system, network administrators and many security staff learnt at the peak of the problems with network worms have been forgotten, and many [all?] systems put in place in those days, disabled. How else can we explain the success of the spread of Conficker, especially variant B or later on the network of many, many organisations and academic sites?

There is a well known saying[13]: "*Those who cannot learn from history are doomed to repeat it.*"

It seems that there are a lot of people out there that need to take heed of this quotation as many threats and attacks borrow from old successful ones.

Which would you rather do? Learn something once, or keep making the same mistakes over and over again?

The latest variant, sometimes called B++[14] adds more functionality and removes the original phone home and auto-update functionality with an improved version, and I don't expect this to be the last variant either. I expect that this year is going to be very busy, again.

Hopefully Microsoft's offer of a bounty on the heads of the authors of this malcode might tempt someone who knows who is behind it, but I wouldn't hold your breath.

Looks like the worm has returned, and this time it means business!

---

[12] Good advice on how to do this can be found here: http://www.publicsafety.gc.ca/prg/em/ccirc/2008/tr08-004-eng.aspx
[13] Attributed to: George Santayana
[14] An indepth technical review of this can be found here: http://mtc.sri.com/Conficker/