FEATURE 1

ZO-TO-BUSINESS

Martin Overton
Independent Researcher, UK

On Monday 15 August something started to spread quickly on the Internet, causing many companies' *Windows 2000* systems to reboot themselves without human assistance. Next, system administrators saw the unexplained slowdown of internal networks. We were once again under attack from a fast-spreading network worm.

MICROSOFT'S UNLUCKY NUMBER?

It appears that, for *Microsoft*, the number 39 is unusually unlucky – at least when it comes to security advisories. Here are three examples of 'the curse of 39' in action:

- MS02-039 – exploited by Slammer
- MS03-039 – exploited by Blaster
- MS05-039 – exploited by Zotob

Each of these worms caused a significant outbreak. In all cases, not only did they cause mass infection very rapidly, but they also had a significant impact on the networks of companies they had infected, in some cases to the point of exhausting all available bandwidth. So, let us now look at the latest 'curse-of-39' worm and see how it fared.

MS05-039 OR BUST!

On 9 August, *Microsoft* released security advisory MS05-039 [1] which revealed a vulnerability in the Plug-and-Play component of *Windows 2000*. The vulnerability was rated as critical. *Microsoft* also released a fix to patch the loophole.

Barely five days after the warning, a worm called Zotob [2] appeared that exploited the loophole. This meant that all those systems which had not been patched, or were not protected by other methods were vulnerable to a dose of digital pox.

According to *F-Secure* [3], Zotob was captured and an initial analysis was made at around 12pm (GMT) on 14 August.

The initial analysis of Zotob.A mentions that the worm may be using the 'houseofdabus' exploit code [3] and that when a system becomes infected it scans the network for other systems via port 445/tcp, at a rate of 300 threads per infected system. Each thread will attempt to connect to a random IP address, created by taking the first two octets of the current system's IP address and randomising the last two octets – e.g. if the system infected has an IP address of 10.10.10.1 then it will attempt to scan random IP addresses in the range 10.10.0.0 to 10.10.255.255.

Any system that shows the port to be open (*Windows 2000* and *XP*) is sent a copy of the exploit code, regardless of whether it has been patched, or is vulnerable.

If the system is an unpatched *Windows* 2000 system, then the exploit code should run and cause a buffer overflow unless the system is protected in other ways. If the exploit code runs successfully, this will create a shell (CMD.EXE) which listens on port 8888/tcp. The scanning (infected) computer will then try to send an FTP script to the newly listening shell on the victim computer. This script is written to the victim's hard disk as '%SYSTEM%\2pac.txt' which tells the newly exploited victim to download a copy of the worm binary from the attacker.

The attacker's FTP server runs on TCP port 33333 and exists only to act as a pickup point for the worm's binary, which is called 'haha.exe'. When run, this downloaded file creates a copy of itself in the %SYSTEM% directory (e.g. C:\WINNT\SYSTEM32 or C:\WINDOWS\SYSTEM32) as a file called 'botzor.exe'. It then creates a mutex named 'B-O-T-Z-O-R' to ensure that only one copy of itself is running on the newly infected system.

Next it adds itself to the system registry to ensure that it is loaded each time the system starts, and also adds a key which disables the shared access service. The newly infected system now connects to IRC server 'diabl0.turkcoders.net' on port 8080, effectively signing in for service as part of a botnet.

Zotob also adds a list of common anti-virus and security-related sites to the hosts file on the newly infected system. This is to try to stop the owner accessing the sites for updates or information. All entries are redirected to 127.0.0.1 (the local loopback address).

Zotob also writes other strings into the hosts file of the newly infected system, these are:

```
Bozor2005 Made By … Greetz to good friend Coder.
Based on HellBot3

MSG to avs: The first who detects this worm will be
the first killed in the next 24 hours!
```

The mention of HellBot3 is a clear indication that Zotob was based on Mytob.

Although Zotob.A can't infect *Windows XP* systems automatically, the worm code can be installed manually or by clicking on an infected file, which will then infect the system running *XP* and Zotob will start scanning for new hosts to infect and exploit. Of course, some of the later variants also spread via email, just like many of the Mytob variants do.

ARRESTED DEVELOPMENT

Several weeks after the initial outbreak of Zotob, breaking news arrived [4], stating that Moroccan authorities working

with the FBI had arrested 18-year-old Farid Essebar, a Moroccan national born in Russia who went by the screen moniker 'Diabl0'. A 21-year-old Turkish citizen named Atilla Ekici, aka 'Coder' was also arrested in Turkey.

The hacker pseudonym 'Diabl0' can be found in around 20 variants of Mytob, which may implicate Essebar as the author. It is also alleged that Mr Essebar was paid by Mr Ekici to create the Zotob worm which Mr Ekici is believed to have distributed. The article also indicates that Essebar and Ekici may have used the information they stole from infected computers to facilitate a bank card forgery scam.

Further breaking news came on 30 August [5], stating that the FBI had confirmed that Turkish law enforcement officials were investigating 16 more suspects in connection with the Zotob worm and its variants. So we may yet see more arrests in relation to Zotob.

THE AFTERMATH

At the time of writing this article, there are 14 variants of the Zotob worm (according to *Trend Micro*), as well as several other worms which use the same exploit to get them onto target systems.

It has been suggested that well over 100 large companies were hit badly by Zotob. These include *CNN*, which provided open coverage of its own massive outbreak. The *New York Times* and *ABC News* were also reported to have suffered from a widespread infection of Zotob. One report suggests that systems the U.S. Department of Homeland Security uses to screen airline passengers entering the United States may have been disabled temporarily by the worm. Other large multinationals reported to have been infected include: *UPS*, *General Electric*, *Caterpillar*, the *Canadian Imperial Bank of Commerce* and *BMO Nesbitt Burns* [6].

MITIGATION

Let us now look at ways in which we could have slowed, hobbled or stopped Zotob in the first place.

Patch me if you can!

Many organisations have patching cycles; each new patch from *Microsoft* is rated individually according to the risk to the relevant infrastructure, and is then tested to ensure that the cure is not worse than any disease that may come along to take advantage of the infection vector the patch mitigates. In most cases this cycle takes a minimum of 14 days, and may be as long as 120 days from analysis to full production deployment.

However, after Blaster, Slammer and Sasser many organisations have pulled the window in to an average of around 30 days. Some organisations now have a 7–10 day patch testing turnaround. But, as we have seen in the case of Zotob, a patch test cycle of 7–10 days is just not fast enough. We can expect other worms to arrive which won't allow any time for patching and which will become widespread as quickly as Slammer, Blaster and Zotob. So, what can be done to offset this risk? The following covers a number of the more obvious solutions that you should already have in place or be considering.

Personal firewalls, network firewalls and routers

As a rule your perimeter firewalls should not have allowed port 445/tcp (and udp) to traverse from/to your network and the Internet. Likewise, if you had set your router ACLs to block traffic destined for systems on port 445/tcp, even if you had Zotob on your network its progress would have been slowed dramatically.

If your systems had personal firewalls installed Zotob would probably have been stopped from scanning your network and infecting other vulnerable hosts. Likewise, if you had a managed personal firewall policy you could have pushed out a new policy to block port 445/tcp inbound (which would have stopped even a vulnerable uninfected system from becoming infected via the port scan) as well as outbound (which would stop an infected system from scanning your network for new victims).

IDS and IPS

As soon as details of Zotob and its spreading pattern emerged, it was a fairly simple matter to create some basic signatures/rules for Snort. These were followed quickly by binary signatures that would trigger on the worm being sent from one system to another, just after it had been exploited via PnP. This was extremely useful as it would list both the attacker's and victim's IP addresses, which would allow faster remediation, or at least removal of the infected systems from the network.

On 11 August, *Sourcefire* had written and released signatures (of high enough quality to be used in an IPS) for the exploit code used in Zotob, and the copycats. IPS signatures for the exploit used by Zotob had been available since before Zotob was first spotted, which would have minimised the likelihood that your vulnerable systems would become infected as the IPS would block the malicious traffic.

Anti-virus

I shouldn't need to say this, but you should ensure that your anti-virus is up to date and that all clients are, by default, requested to check for new updates at least once a day. Again, if you have a managed anti-virus infrastructure this can be significantly easier as you can force all connected managed clients to update themselves when an outbreak is

in progress. This will help to shrink the 'possible infection pool' and make cleanup less expensive.

There are a number of other methods which could have been used to mitigate the threat of Zotob (and most other malware), but I have run out of space to describe them.

ZOTOB'S PROGRESS

Finally, the following is a timeline charting Zotob's progress [7]:

9 August 2005: *Microsoft* releases six security patches (MS05-038–43). Four are rated as critical. Initial exploit code is written and released for two of the vulnerabilities; MS05-038 and MS05-041.

11 August 2005: Exploit code is written and released to take advantage of the vulnerability patched in MS05-039. This is the PnP (Plug and Play) vulnerability.

12 August 2005: Snort signatures are released to detect the exploits, and code for another MS05-039 exploit is released.

14 August 2005: A new worm based on Mytob code and containing exploit code as its attack vector is released, discovered by *F-Secure*, and named Zotob. The exploit code used is from the 'houseofdaubus' hacking group (exploit code from the same group was used in the Sasser worm).

15 August 2005: The source code for the widespread IRCbot family is updated to take advantage of the MS05-039 exploit. New variants of Zotob appear. Snort signatures for detecting the binary as well as the IRC traffic are written and released. Most anti-virus products can now detect Zotob.A.

17 August 2005: There are now seven variations of Zotob, one Rbot, one SDbot, one CodBot, three IRCbots and two Bozori variants using the PnP vulnerability. The Bozori and IRCbots are deleting other bots. The Bot-wars have begun!

So there you have it, Zotob in a nutshell.

REFERENCES

[1] http://www.microsoft.com/technet/security/bulletin/MS05-039.mspx.

[2] http://www.microsoft.com/security/incident/zotob.mspx.

[3] http://www.f-secure.com/weblog/archives/archive-082005.html#00000624.

[4] http://blogs.washingtonpost.com/securityfix/2005/08/arrest_of_zotob.html.

[5] http://www.computerworld.com/securitytopics/security/story/0,10801,104269,00.html?from=story_kc.

[6] http://business.timesonline.co.uk/article/0,,9075-1738986,00.html.

[7] http://singe.rucus.net/blog/archives/510-MS05-039-and-the-Zotob-summary.html.