Yui Kee Computing Ltd.

Newsletter

November 2001

Contents

AVAR 2001 Conference

AVAR 2001 is a rare opportunity to meet and hear the industry's top anti-virus experts. The line-up is headed by Dr. Vesselin Bontchev, voted the individual who contributed the most to the anti-virus industry at the Virus Bulletin Conference 2000, as the Day 1 Keynote Speaker; and Ms Eva Chiang, who received the Secure Computing Lifetime Achievement Award in April 2001, as the Day 2 Keynote Speaker.

The Fourth annual Anti-Virus Asia Researcher's Conference will be held 4-5 December in Hong Kong. Full registration details at the website:

http://www.aavar.org/avar2001/home.html

Incident Update: Learn from Badtrans

For the past four months, W32/Sircam@mm has topped Messagelabs' list of most active viruses, but W32/Badtrans.B@mm finally toppled it over the weekend.

W32/Badtrans.B@mm first appeared on the 23 November and spread rapidly among home users in the UK on Saturday and Sunday. On Monday morning the pattern of victims changed to business users as people started work. People opening their email before their anti-virus software was updated to catch Badtrans.B may have assisted this. When executed, it emails itself to many addresses as an attachment. The attachment may have different names, but it will always have a double-extension. The worm also drops a password stealing Trojan (if your site has been infected, force users to change their passwords as a precaution). If the recipient is using some versions of Microsoft Outlook, the infected attachment will execute automatically if the message is viewed or previewed. A security patch from Microsoft released on the 29th March 2001 fixes the bug that allows this to happen.

The rapid spread of Badtrans.B was avoidable:

i) Attachments with double-extensions should be blocked at the mail gateway - such files are automatically suspicious (some Unix users might like to send and receive .tar.gz files, this can easily be accommodated with a slightly more complex rule).

ii) Software should be kept up-to-date with the security patches. Make sure the patches get re-applied after re-installations, e.g. when hard disks crash, by having a "standard

installation checklist".

iii) Automate your anti-virus updates, consider the relationship between the update schedule and user activities: e.g. schedule an update before users open their mailboxes on Monday morning. Of course, systems that run all the time, such as mail servers and gateways, should have updates scheduled without regard for weekends and holidays.

iv) Instruct users to follow Safe Hex.

Any one of these would have slowed the spread of W32/Badtrans.B@mm, and many other viruses, all of them together would be very effective. Why are people not doing the simple things to protect themselves?

Full Disclosure

The debate on full disclosure has been heating up recently, with Scot Culp of Microsoft accusing the security community of "Information Anarchy" and a group of companies, including Microsoft, planning to announce their own guidelines restricting disclosure. Full disclosure is the principle that, when a researcher finds a security hole, he or she should promptly publicise it widely. Supporters of full disclosure claim it improves the security of products because vendors are forced to fix the holes while critics point out that the bad guys have an easy opportunity to attack before a fix is available.

Scot Culp is clearly against full disclosure, claiming, "The relationship between information anarchy and the recent spate of worms is undeniable" in an article published on the Microsoft website. He points to close similarities between recent worms and published exploits as evidence that the worm's authors benefited from the disclosure. However, some in the security community have pointed out that, at least in the case of CodeRed, the worm was not based on the disclosure by eEye, but on an earlier worm that did not become widespread.

Unfortunately, commercial concerns also affect the viewpoints. A small security company that discovers a flaw in a major product can get massive publicity by being first to publicise it in the popular media. Conversely, the vendor of the major product suffers negative publicity; it would be better for them if the flaw remained unknown until the product became obsolete. Failing this, they can minimise the effect by announcing their fix at the same time as the flaw. The reality of the bad publicity effect can be seen in Gartner's advice to dump IIS, although it remains to be seen whether significant numbers of IIS servers will be replaced.

However, the real concern is the threat to the users, and the key here is that the vulnerability existed before it was discovered. Indeed, the company or person who first publicises the flaw might not have been the first to discover it - an unknown number of bad guys might already be using it. A smart bad guy who discovered an unpublicised flaw could maximise his gain from it by quietly using it for his nefarious purposes, and then releasing an automated attack tool as soon as it becomes publicised. It would then appear that the announcement resulted in the creation of the tool, discouraging future disclosures.

A security vulnerability opens a window of opportunity with three distinct phases. The vulnerability exists as soon as the product is released, but the window does not open until it is discovered. If a bad guy discovers it, the first phase begins; this is where the vulnerability is exploited without the possibility of response. The second phase begins when the vulnerability is publicised. At this stage, the defenders can take action, even though a fix is not yet available. The action will depend on the nature of the vulnerability and the defender, for example, if sufficient information is available, the defender might program their intrusion detection software to shut down the vulnerable server if the attack is detected - choosing denial of service in preference to theft of corporate secrets. The third phase starts when the fix is published - defenders can then start to eliminate the vulnerability. The third phase ends, and the window closes, when all vulnerable systems are eliminated - in some cases, this might be when

everyone has stopped using the product. Alongside these phases, but semi-independent, is the publication of an automated attack tool exploiting the vulnerability. Before publication of such a tool, only skilled attackers can exploit the vulnerability, after publication, any script-kiddie can attack. The publication of the tool might mark the beginning of phase two - it could be the first public announcement of the vulnerability, or a tool might never be created.

The times when the vulnerability can result in the most damage are in phase one, when the attackers can choose valuable targets with impunity, and phase two after the publication of an automated attack tool (I will call this phase 2b), when the number of potential attackers skyrockets. Full disclosure advocates seek to minimise phase one, and their opponents seek to minimise phase 2b. The conflict arises because the actions that defenders can take to minimise one of these results in lengthening the other. Any description of the vulnerability will give a skilled attacker enough information to investigate the vulnerability him- or herself, allowing the production of an automated attack tool. However, any delay in the publication of a description extends phase one.

To some extent, the debate over full disclosure is irrelevant when the most damage in the real world happens during phase three - after the patch has been publicised. The most recent example is W32/Badtrans.B@mm showing that, even when a patch is available, it often is not applied. This is particularly true for mass-market, consumer-orientated software, which is installed and maintained by people with minimal technical skills.

Personally, I am moderately in favour of full disclosure - we cannot wait for security solutions to be handed down from On-High, in accordance with vendor's internal timetables and interests. This should be combined with a responsible attitude - certainly, those who can assist in the defence (the vendor, and perhaps others, such as intrusion detection developers) should be informed in confidence immediately, with full details. A public announcement should be scheduled without regard for the vendor's preferences. Some researchers may delay as little as one day before a public announcement; CERT/CC's policy is 45 days delay, with the possibility of variation according to the nature of the vulnerability. I agree that different vulnerabilities will be best dealt with by different delays. The public announcement should contain all information that would help defend against attack, and as little detail of how to launch an attack as possible. However, if the vendor claims the attack is infeasible or unimportant, the researcher should back-up the announcement by demonstration. This could be a non-automated tool, distributed only to trusted researchers and technical journalists.

Thus, I think that, in the case of CodeRed, eEye went too far in publishing the details of forceful heap violation. Some extreme full disclosure advocates even publish the source code of viruses - this is an unacceptable release of an automated attack tool. They are also missing the point - the real vulnerability that viruses exploit is that we are using general-purpose, programmable computers; computers are vulnerable to viruses because they are useful and adaptable.

On the other side, Scot Culp is going too far in demanding that researchers wait indefinitely for vendor patches. He says that Microsoft will be working to build an industry-wide consensus on this issue, I welcome that, and I hope the consensus will represent the best interests of the users, not the commercial interests of vendors or security companies.

YUI KEE
銳基電腦顧問有限公司
Anti-Virus, Network Security & Education

Suite C & D, 8/F, Yally Industrial Building
6 Yip Fat Street, Wong Chuk Hang, Hong Kong
Tel: 2555 0209 Fax: 28736164
E-mail: info@yuikee.com.hk
http://www.yuikee.com.hk/computer/