Global Information Assurance Certification Paper

The Bubbleboy who Cried Wolf: a Retrospective

By Chuck Rothman

The headlines were alarming:

"Virus Called Biggest Computer Threat"[1]
"A Virus Spreads Without Opening Any E-Mail Files"[2]
"Virus Puts U.S. On The Alert — Seinfeld'-Themed Bug Could Be Granddaddy To Vicious Strain"[3]

This was the Bubbleboy virus, called by experts, "the next evolution in viruses.[4] It was the first virus that could execute automatically from e-mail, without the user having to click on the icon or do anything other than read the file – "Good Times" for real.

Over a year has passed. There have been many major virus attacks, most notably Ilove you, Navidad, Hybris, and MTX. But no Bubbleboy. It not on any list of most common viruses. More importantly, none of the common viruses have Bubbleboy's "next evolution" trait of spreading without user action. VBS viruses are the big growth area, but they all use social engineering concepts to entice the user to open them. They do not run automatically. One would think that a virus that spreads merely by reading an e-mail would be something virus designers would love to create. Yet none have.

So why no Bubbleboy? Why isn't the model commonly used? Is it even used at all? Why all the warnings? This paper will discuss the virus and try to answer these questions, and well as assessing the implications of Bubbleboy – and the reaction to it – in the computer security industry.

What Bubbleboy Does.

Bubbleboy is a VBS virus, one that uses the scripting capabilities of Microsoft Outlook and Outlook Express to spread. When an e-mail containing the virus reaches a machine running Outlook Express and MSIE 5.0 and using the preview pane, it automatically executes.

The payload was relatively benign. A file, update.hta, is written and then run. This makes changes to the registry (changing the name of the owner to "Bubbleboy," changing the organization to "Vandelay Industries") and then sends itself out as an email to every name in every Outlook address book.[5] This has the same potential for trouble that the Melissa virus did – it can overload a mail server very quickly.

[1] The Toronto Star , November 11, 1999
[2] The Wall Street Journal, November 11, 1999
[3] Daily News, November 11, 1999
[4] Sal Viveros, Network Associates press release.
[5] McAfee Antivirus Encyclopedia Online, http://vil.mcafee.com/dispVirus.asp?virus_k=10418&

How does it manage this without user intervention? All sources mention one important fact that was ignored in the news stories. Bubbleboy exploits known security vulnerabilities in MISE and Outlook.

Microsoft was aware of this problem well before Bubbleboy existed. It was written up in their security bulletin MS99-032, "Patch Available for 'scriptlet.typelib/Eyedog' Vulnerability."[6] The bulletin outlines the problem with scripting and explains how two ActiveX controls – scriptlet.typelib and eyedog – were inadvertently sent out as "safe for scripting." This meant that Internet Explorer would be able to run these controls without warning the user. Since scriptlet.typelib allows local files to be altered, this gave writers of malicious code an opportunity for mischief.

The security bulletin warned about the vulnerability, but gave a simple remedy – a patch that fixed both controls so that they were no longer marked "safe for scripting." It would seem likely that, since this was an inadvertent mistake, updated versions of Outlook Express would not have the problem.

One important thing to note is the date of the bulletin. It was originally released August 31, 1999, with updates on September 2 and October 12.

The Bubbleboy warnings began to appear on November 11, one month after the last update.

Warning! Warning! Danger! Danger!

Once the Bubbleboy warning was sent out, major newspapers took up the cry. Articles appeared in *The Boston Globe, The Daily News (New York), The New York Times, The Toronto Star, The Wall Street Journal, The Washington Post, The Atlanta Journal and Constitution*, and *the Los Angeles Times*.[7] All stressed the angle that this was a "new type of virus."

But was it?

Bubbleboy may have been new in one sense – it did run from e-mail without user intervention – but in many others, it was a case of seeing it all before. It is hardly the first bit of malware that exploited flaws in program code. In many ways, it's similar to exploits like the null session vulnerability in Windows NT, buffer overflows, or SYN/FIN attacks. Bubbleboy was merely another in the list of exploits that find a vulnerability and design a way to take advantage of it.

There is one thing that makes Bubbleboy different from many exploits. Usually, it is hackers who discover the vulnerability and make use of it. In this case, the vulnerability was discovered and announced by the *software maker*, well before the virus was seen. It seems likely that the virus creator decided to create the virus solely to exploit a known

[6] Microsoft Technet, http://www.microsoft.com/technet/security/bulletin/ms99-032.asp
[7] All found at http://www.lexis-nexis.com/universe (subscription site)

security weakness, and one that was inadvertently part of the software. That is unusual for a virus (the weaknesses in Microsoft Outlook, for example, are generally due to the VBS scripting security, but that is a deliberate decision by Redmond to use that capability, not due to a true flaw in programming).

How widespread was Bubbleboy?

It appears at the time of warnings, Bubbleboy was rare. It did not appear on the wildlist for November 1999.[8] It began to show up the next month, but with only three people reporting it, a number that stayed stable (with the same three people) for the next several months. Not exactly an epidemic, especially compared to unremarked viruses W32/mypics (4 reports), W95/Babylonia (7), and W97/Turn.a (4) – all first seen on the wildlist the same month as Bubbleboy. The VBS_Freelink virus – which attracted no headlines – was reported by twelve of the Wildlist's reporters in November, one month after it was discovered. That's three times the number of Bubbleboy sightings, but not one talks about VBS_Freelink.

Bubbleboy certainly had some theoretical potential for greater spread, but it just never did. And even with that potential, McAfee still classifies Bubbleboy as "low risk."[9] In any case, reports of infection are rare.

In addition, the Sophos Antivirus site has the following entry in their Bubbleboy FAQs: "This virus is not considered to be in the wild. It appears to be a "proof-of-concept" and the virus's author has sent his creation to various anti-virus companies to taunt them with what he has achieved."[10]

So why the hype over a low-risk threat? The answer may lie in the nature of the anti-virus industry. This was not the first time anti-virus vendors made dire predictions about a dangerous and widespread virus, which turned out to be overblown. Antivirus vendors need to sell their software, and a new and dangerous virus is just the thing to keep them in the limelight. The articles in the newspapers all quote from one or two sources, which indicates they were just reporting press releases they had received.

But even at the time the word of Bubbleboy was going around, antivirus vendors were under criticism for their warnings. According to Graham Cluely, senior technology consultant at Sophos Antivirus:

"Some people are doing the industry a disservice. There is a problem with hype. There are around 48,000 viruses that we know of. A couple of hundred are in the wild. BubbleBoy is not one of them. It is safe inside laboratories."[11]

---

[8] Wildlist, http://www.wildlist.org/WildList/

[9] McAfee Virus Encyclopeida online, http://vil.mcafee.com/dispVirus.asp?virus_k=10418& Bubbleboy was considered low risk by McAfee even back in November 1999, at the same time they were spreading the word about Bubbleboy's danger.

[10] http://www.sophos.com/virusinfo/articles/bubbleboy.html

[11] Uhlig, Robert, "Connected: BubbleBoy myth burst," Daily Telegraph (London), November 18, 1999, p.2.

Has Bubbleboy spread? It doesn't seem so. And, interestingly, it has not spawned any variations. Any successful virus is quickly modified into slightly different forms in order to attempt to fool antivirus software looking for a particular combination. Thus, a virus like Melissa spawns Melissa.A, Melissa.B, etc. Bubbleboy, on the other hand, has shown few variants (McAfee, always quick to add them, has one listed, but other antivirus vendors list none at all), a sign that it is of little interest to the people who write and spread viruses.

Why not Bubbleboy?

Why didn't Bubbleboy be a standard model for spreading viruses? It certainly would allow for the rapid spread, an ideal situation for anyone creating a virus.

There are several obvious reasons. The simplest is that because it merely exploited a known security hole, and one that was easily closed, it wasn't worth anyone's time to design a virus to spread by this method. The software vulnerable to this sort of infection was limited. Sure, Outlook Express is widespread, but the virus only worked on one specific version. A simple patch eliminated the vulnerability, and, as Microsoft rolled out newer versions of the software, the number of vulnerable machines decreased even faster. This isn't like other security holes; the nature of computing ensured that more and more MSIE users would upgrade to newer versions. At this point, MSIE 5.5 is current – and free. This makes upgrading simple and inevitable. People who have no virus protection at all will protect themselves from Bubbleboy as a matter of course, and the number of vulnerable machine will soon become too low to support any major outbreak.

But someone could have come out with something quickly, a Melissa-like virus that would take advantage of the security breach and spread like crazy. It didn't happen. The closest thing to this scenario involved the "Iloveyou" virus, which, like everything else, depended on user intervention to be activated.

It's impossible to know the answer for sure. An interesting speculation, though, may be in the nature of those who write viruses. They love to exploit the design weaknesses of software, and it's a special bonus if they can find these weaknesses themselves. But the scriptlet.typelib/Eyedog vulnerability was not discovered by a hacker. It was discovered and announced by Microsoft. Call it "reverse social engineering": a virus writer is unlikely to want Microsoft – the "evil empire" and major target of hackers – to do their work for them. The ego payoff was too small.

So What's the Point?

So why look at the Bubbleboy virus at this point? Because there are important lessons to be learned from it.

- *Never say anything is impossible*. Virus writers are always making breakthroughs and looking for loopholes. Though Bubbleboy seems to be created as an exercise, it did show how a weakness could be exploited. Any general rule – in this case,

"you can't get a virus merely be reading e-mail" – can be broken given the correct set of circumstances and vulnerabilities.

- *Virus warnings have to be researched*. A look into how and what Bubbleboy was could have prevented panicked articles in the newspapers. The virus was merely a curiosity, but the public was told that it was one of the most dangerous viruses ever (potentially, of course). Of course, the newspapers aren't computer security professionals and are prone to accept whatever an antivirus vendor tells them. However, professionals in the field should be more careful. A look at the facts at the time would have made it clear that Bubbleboy was a minor curiosity that could be prevented by a few simple steps.

- *Be careful of what antivirus vendors tell you*. Their job, after all, is to sell software, and the more attention they get, the better. Even a rare virus can be an opportunity to get headlines, especially if you can make the case that it is something new and different. Even an old virus can result in a warning that gets spread over the media, like the recent "revival" of the AOL Password stealer warning.[12]

- *Even legitimate threats can be overblown*. Paranoia is no substitute for a detailed analysis of the threat and a calm reaction to it.

- *There are long-term effects of any hysteria*. Now, people – even security professionals – think that there is a serious threat of viruses that can that can infect computers from the Outlook preview.[13] They are not wrong to say it *could* happen, but they are remiss in implying it is *likely* to happen.

- *Crying wolf is going to make people distrust your messages*. That may be harmless when the virus is harmless, but when a truly dangerous virus is unleashed, it'll be harder to get people to react to the danger. It will be "just another virus warning."

The fact that no new virus since Bubbleboy has used this method is an indication that this "new class of virus" probably has only one member. Bubbleboy itself was an interesting exploit of a security breach, but is hardly one of the top viruses. The meaning of Bubbleboy is clear: don't trust what you see in the papers.

The final problem of Bubbleboy and its like are best summarized by Dan Schrader, the chief security analyst at Trend Micro: "Anti-virus companies have always been seen as ambulance chasers, and sometimes, it's true. Because this is an industry that has been built on hype and alerts and pretensions of being good citizens, the industry

[12] Shiver, Jr., Jube, "McAfee Issues Controversial Bug Advisory", *Los Angeles Times, February 2, 2001,* http://www.latimes.com/class/employ/showbiz/20010202/t000009820.html. McAfee claim is saw a 100% increase in the trojan, but that could mean that the number of instances jumped from two to four.
[13] SANS NewsBites Vol. 3 Num. 01. The editor calls this a "fundamental shift in viruses," spreading the misconception.

doesn't have a lot of credibility.[14]" As professionals, our job is to clearly and calmly explain the dangers of a particular threat, without resorting to hype or overreaction. The reaction and coverage of Bubbleboy does the profession a disservice.

[14] Quoted in Dean, Katie, "The Virus Ambulance Chasers," *Wired News,* May 19, 2000. http://www.wired.com/news/technology/0,1282,36464,00.html

Last Updated: May 25th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event