Global Information Assurance Certification Paper

GIAC Security Essentials Certification (GSEC)
Version 1.4b (Option 1)

Diffusing a Logic Bomb

Nicolas Robillard
January 2004

Abstract
This document covers the definition of a logic bomb, why and by whom they are used, how they work, how to limit the destruction capabilities of such bombs and how to diffuse them. This paper will also relate some real life cases where logic bombs were used.

What is a logic bomb?

When talking about computer threat, most people think about hackers and viruses. Logic bombs rarely come up in discussions mostly because the concept is often associated with viruses and/or trojan horses. First, let's define what is a logic bomb, also known as slag code. A good description has been written by David Stone, University Laboratory High School: "A logic bomb is computer instruction that codes for a malicious act when certain criteria are met, such as a specified time in a computer's internal clock or a particular action, such as deletion of a program or file." (David Stone, http://lrs.ed.uiuc.edu/wp/crime/viruses.htm). A more complete definition can be found at the information security glossary (http://www.yourwindow.to/information-security/gl_logicbomb.htm.)

A virus can act as a logic bomb if by example the virus waits until a specific date to run its destructive payload (like the WM/Theatre.A which destroy the system hard drive if the system clock is set to the first of any month, see the link in the references). In fact, most destructive payload can probably fit the description of a logic bomb but not every virus includes a destructive payload since many viruses only replicate themselves. The same conclusion can be made about logic bombs and trojan horses. But the fact is that a logic bomb can exist by itself without replicating itself (acts as a virus) or opening a backdoor (acts as a trojan). To complicate it even more, it can also be attached as a legitimate file. This paper will focus on the concept of a logic bomb whether carried in a virus, trojan horse, legitimate file, etc.

Why is it used and by whom?

The 2003 CSI/FBI Computer Crime And Security Survey reports that disgruntled employees is the second likely source of attacks (77%), preceded only by hackers (82%). So, how uses logic bomb? As discussed earlier, viruses sometime include slag code. Hackers can put a logic bomb on a compromised

system and set it to destroy the system if its newly installed backdoor ever gets uninstalled, an unhappy employee can put a logic bomb on the system before getting fired or because he didn't get a promotion that he thinks he deserves, etc. In fact, anybody with malicious intent is susceptible to use one.

From 1980 to 1985, some software developers imbedded logic bomb into their software, set to destroy the software itself if the license was not renewed. Of course, today this practice is illegal, but people are still using logic bombs in other contexts to achieve their ends.

The urban legend of the "salami scam" also relate the use of a logic bomb. Like in Superman III, the scam is based on logic bombs that redirect a couple of cents (or even fractions of a cent) in an account owned by the programmer. This is considered as a fable because even if you can find tons of stories involving the salami scam on the Internet, there's no serious and proven case to this date. But even if today there's no proven case, it doesn't mean it will never happen. In fact, many known cases of fraud were executed using logic bombs.

This document will put more emphasis on the disgruntled employees, because hackers and viruses/worms tend to attack generic vulnerabilities like common or weak password to successfully hack/infect the largest number of hosts. Even if they use a more destructive approach like destroying every data on the system or file corruption, these activities can be tracked with an integrity checkers like tripwire and the system can be restored with backups. On the other hand, an employee with the password to the HR database can make legitimate changes to the database (actually the change are not legitimate since they are not approved, but the employees use a legitimate access) that can be found only weeks later. You would have to manually verify every entry in the database made since or restore the database to the last good state and re-enter every entry that was added since. The point is that an employee already knows good exploitable information and can hit your system harder than any hacker or virus could do.

Not every employee can make a logic bomb. Even if some are available on the Internet, most of the time some programming skills are required, or at least some scripting skills. The employees also need to have proper access to implement this type of code. Most of your IT staff should fit this profile, but other non-IT employees may also have the qualifications and access.

How does it work?

The definition of a logic bomb stipulated in the first section of this paper can be divided in two distinct parts: the triggering and the payload.

The Triggering:
The triggering is what relates this type of code to a real bomb. When setting up a bomb you would like to have some time to run away before it explodes. This could be done by setting a timer or by sending a radio signal to the bomb when you're out of danger. The same principle applies to a logic bomb. You may plant a slag code somewhere in the financial system and tell it to "explode" 6 months from the current date. If you're still an employee at this time you can add another 6 months to the counter, but if you've been fired the destructive payload will be unleashed. Here is a list of possible triggers that exist. This list is not exhaustive, but the most common triggers are listed here:

- Specific date/time: The payload will be executed when the system clock is equal or higher than the specified date/time.
- Countdown: Work like the specific date/time trigger but does not rely on the system clock. Instead the trigger implements its own timer bases on the number of seconds elapsed since the logic bomb was activated. This is somewhat harder to implement, but also harder to diffuse. Most slag codes relying on a specific date/time trigger can be tricked by changing the internal clock of the system. This trick doesn't really diffuse the bomb but can buy you some time to find a way to actually disable it. In this case, changing the internal clock won't alter the timer implemented into the slag code, so it won't do any good.
- Third party triggering: A time bomb can also be triggered by an external scheduler software like Windows Scheduler or the Linux Cron. These kinds of slag codes are particularly easy to make since the triggering code is already made for you. You can, by example, create a batch file that formats the system drive and schedules it to run in a few weeks.
- Reset: This trigger must be combined with one of the first three triggers. It's simply a way to extend the time before the bomb actually goes off. In the movie Safe House (see the URL in the references), an ex-government agent played by Patrick Stewart has life threatening information about his former employer. He puts in place a time bomb that sends all the information to different magazines and newspaper if he doesn't enter the correct password each day.
- State Changing: This one checks for changes on a specific entity before running the payload. Is my name still in the HR database? Is my account still active? … Etc. It can monitor everything, to registry keys, passwords files, database entry, system configuration, etc. This type of attack can be launched from outside the company if the programmer manages to check an external web site for state changing.

We can also imagine other triggering devices such as a key logger that executes the payload when a specific sequence of keys is typed. The disgruntled employee imagination is the only limit to what will be the trigger so be ready for everything.

The Payload:
The payload of a logic bomb is the destructive part of the code. It could be as simple as **format c: /autotest** (this feature is no longer available on windows 2000 and up) or more subtle like removing or modifying some database entry that will not be discovered for a long time. Remember that your employees have a very good knowledge of your working environment and that every piece of information they have access are a potential target. Also remember that the payload will be executed with the privileges of the account used to run the logic bomb. This means that a user can actually have the rights to activate his logic bomb but may not have sufficient accesses to delete system files. This points the fact that an employee will only target what he has access to. It's important to remember this while searching for a logic bomb on the system. Don't lose time scavenging the part of the system that the employees have no access before having checking out every place they actually have access.

Now with a counter example: lets say that your company has a computer park of over 5,000 computers. Your company will most likely use some sort of automated installation method for its computers like Norton ghost. The local administrator account will be the same on every PC of your company. How hard would it be for an employee to get the local administrator password for its machine? With tools like LC4 and Advanced NT Security Explorer it's a matter of seconds. Now the employee can erase most data on any computer of your enterprise, make them reboot, crash, etc. He would have total control over those machines. Now how long would it take to get your company up and running again, if a logic bomb completely destroyed all data on every PC of your park? What about all the files residing on your employees computers that were never backed up? Even if your company has specific policies about not saving important data on the local drive, experience tells us that if the user can, he will probably do it.

This example was to point out some important facts.

- When we speak about the access that a specific user has, we must also include all accesses he can grant himself by some other means.
- The employees will not necessarily target critical systems like the HR database. Some systems with less importance can do as much damage to your enterprise it attacked altogether. As another example we can also think of a bomb that could change the configuration of every network equipments, blocking all communications between hosts.

Now lets add a bit more of complexities to the problem. As it will be mentioned in the prevention section, some softwares, like anti-viruses, have the ability to stop the execution of a program following certain behaviors (like erasing system files, etc). Other program like SecureEXE will only allow the execution of programs

approved by the system administrator. In that case, a logic bomb should not be able to run (unless the employee convinces the system administrator to trust his program). But as seen in the triggering section, it's possible to use third party software as a trigger. Is it possible to do the same for the payload? Absolutely! What about scheduling a destructive SQL Query in argument to your favorite SQL command interpreter through the windows job scheduler? We just turn two innocuous programs in a destructive logic bomb. Worst, the system administrator approved those two programs.

Prevention:
There's no way to completely prevent the use of logic bombs in your system. But there are way to make their life harder:

- Use the concept of Least Privilege. Don't give too much power to your users, only what they need and review their rights regularly. This will ensure to limit the range of potential targets a specific user can attack.
- Stay up to date. If the user doesn't have sufficient access he may try to get it through a privilege escalation technique. If you patch your system regularly, this could prove to be much more difficult.
- Use secure system configuration. Hardening guide can be found on the net for most platforms. Also make sure to use a different password for each account on different host.
- Make a baseline of known processes running at any given time on every host. Compare the baseline with the current view regularly. This will help you find rogue process on your system.
- Use an integrity checker like "Tripwire" to find if any software has been modified to include a logic bomb in it.
- Check your scheduler to make sure that no unknown jobs are scheduled.
- Review the log for pattern, or strange behavior.
- Be sure to protect every hosts (workstations and servers) with an up-to-date anti-virus that has the ability to detect and block known "destructive programs" as well as unidentified ones with heuristic and pattern recognition.

These prevention techniques don't only apply to logic bombs, but will also be useful to prevent from hackers, rootkits, trojans, abuses of the system, etc. In fact, most of them should already be implemented in your company.

Good guides on these prevention techniques can be found on the net.

Diffusing a Logic Bomb
Lets now face the worst case scenario: a logic bomb was successfully inserted into your system and exploded a few minutes ago. It is time to diffuse the bomb. I

divided the whole diffusion process into six steps. This is only a suggested course of action. This method may or may not apply to any particular situation.

Note: Since we only cover the slag code this is not embedded into a virus, we can take in consideration that there's only one source of "infection" and that it will not replicate. This discussion will only cover the case where only one system is "infected" with a logic bomb, but even if it doesn't replicate, the author of the bomb could have installed it on many hosts. In that case, you would have to repeat the whole process that many times.

1 – Evacuate the area
The first thing to do, obviously, should be finding the bomb. But there's a more important step. Remember that the logic bomb could reside anywhere on your network, not only on the affected host. So it could still do some damage. Since the only thing we know for sure is that we got one affected host, it's crucial to remove this host from the network. Then a careful review of the different logs should indicate where the attack started. If the investigation takes too much time and other hosts are being affected, it might be appropriate to disconnect the whole network or at least the part where critical data resides.

2 – Keep up the evidence
When you successfully find the host, make sure to remove it from the network. Then make a backup of the data for forensic investigation later on. This step is crucial if you intend to press charge against the offender or if you want to understand how it happened.

3 – Restore the data (Optional)
Depending on your backup scheme and the criticality of the data, it may not be an option to restore the last backup of this host. If you need to clean it manually, go to step 5. The best course of action would be to re-install everything from scratch because your are sure to completely resolve the problem, but that may prove to be difficult on a production server. So restoring the environment may be a good alternative.

4 – Verifying the backup (Optional)
This item is optional since it only applies if you restore your system. Once it is done, do not put the host back online. You must first verify that the bomb is actually gone. Remember, the bomb might have been there for a long time, so you may have just restored it as well as the rest. If you also restore the trigger, the bomb might go off immediately (ex: if you restore a time bomb, it will go off because the system time will be greater than the detonation time). There are cases where the trigger will be reset (ex: if you restore a counter bomb that was set to explode 60 days after the program is first run, you may have to wait another two months before realizing the bomb was restored). In any case, if you find that you restored the bomb, you have to get back to step 3.

5 – Diffusing the bomb
If you find yourself in a situation where you must diffuse and remove the logic bomb from the host, make sure to restore the affected system in a lab environment were you can proceed with different tests that could potentially destroy the system.

- Play with the system time. If you're dealing with a time bomb, it will surely go off. You can then set the clock a few days before the incident to make sure it won't explode again while trying to diffuse it. Don't set the clock back with a too big difference. It's easy to program the bomb to compare the system time with is own file timestamp. If the timestamp is higher than the current system time, the bomb might be programmed to explode.
- Install packets sniffers on the machine. If the bomb tries to contact a remote host for any reason (maybe the triggering is caused when a condition is met on a remote machine, or maybe the payload is set to execute on a remote machine), you may find the process associated with that communication and find the executable of the slag code.
- Analyze the logs. The system logs are full of juicy information so don't let any annoying detail without doing a good and valid checkup.
- Check every running process, any job scheduled and any program run at start time. The bomb must be started somewhere, so carefully analyze each entry, verify the integrity of each file by comparing them with a known clean version of it.
- Finally, in doubt, make use of a forensic expert to find and eliminate the slag code.

6 – Restore the service
At this point, it's safe to plug back the system into the network. It's probably a good idea to keep a vigilant eye on the system for the next few months to be sure you didn't miss anything while removing the bomb.

Threat Level
We have covered a lot of theory in the document. But is the threat strong enough to put in place preventive action against logic bombs? The answer is really up to you. Statistics prove that most company will suffer from disgruntled employees abuses, and as shown before, most preventive actions are good against a lot of other threats and should be put in place anyway. The fact is that when a company is targeted by one of is own employee, it won't tend to tell the whole world about it. In fact, there's almost a taboo surrounding most incidents. But some companies break the wall of silence and speak of such incidents to educate the world. Here are some examples:

March 4th 2002, "*A disgruntled former UBS PaineWebber computer systems administrator attempted to profit after detonating a "logic bomb" program that*

caused more than $3 million in damage to the brokerage's computer network, …" Full story at http://www.cbsnews.com/stories/2002/12/18/tech/main533450.shtml

The next two examples are taken from dyslexia, an Internet site dedicated to report all computer related incidents. More examples can be found at the URL listed in the references.

January 12th 1988, "*The trial was billed as the UK's first "logic bomb" case, with McMahon accused of planting unauthorized code in the DEC PDP 11 system software of air freight forwarder Pandair Freight. The prosecution claimed that one such "logic bomb" locked terminals at Pandair's Heston office, near Heathrow, and a second was set to wipe the memory of the company's Birmingham computer.*

McMahon's motive was either financial gain or revenge after losing a 50,000 pound contract with Pandair, the prosecution said." The full report can be found at http://md.hudora.de/blog/guids/99/02/00312301322319529328.html.

Friday the 13th, November 1987, "*A disgruntled employee of a London Ontario company recently planted a surprise in the corporation's computer - a ``logic bomb,'' which would, on a certain date, knock out the entire system. It was found in time and the man was prosecuted ``but it would have destroyed their complete computer system - it would have been down for months,'' Sergeant Ted Green of the Ontario Provincial Police said yesterday*". This story and some other cases of logic bombs are related at the following URL: http://md.hudora.de/blog/guids/26/13/200312301139518780144.html

As you can see, the threat is real and the damage can be costly! The net is filled with this kind of horror stories and since people with good computer skills begin to be more common, this kind of attack will surely grow in the near future. And even a good defense-in-depth can't stop it from happening (Especially when the disgruntled employee is the one who implemented or conceived the defense-in-depth in question).

Foreseeing the Future

Is the future of the computer age a dark one ? Well, the whole concept of computer security has been emerging in the last few years, making programmers realize that it may be worthwhile to put extra efforts to make their programs secure. Windows 2003, IIS6 and all new Microsoft products now come hardened by default. New Linux distributions tend to do the same. Software like IDS, Firewall, Anti-Virus, Integrity Checker, etc, are considered to be the baseline for a good security. In overall, systems seem to become more secure, making them more difficult to attack.

But the dark side is always lurking. The time between a security hole is found and the time an exploit is actually released tend to diminish. Automatic tools created a whole new generation of hackers named script kiddies. Anyone good enough to search the web to find such tools, download it and click the start button, fits this category. The number of automated attacks has gone rocket high. Kids now learn to use a computer at age 3. So malware product (such as a logic bomb) become more widely available each day, they become more easy to use and people are getting better at using them.

The question is: Will the evolution of computer security will surpass the growth in the hacker's community ? As in the movie "Star Wars", when Darth Vader tells his son Luke Skywalker that he is his father, the hackers are probably the fathers of computer security. After all, without a dark side, why would we need protection? Without an action, there's no need for a reaction.

So computer security will probably be a step behind the attackers for another couple of years, always patching what was discovered earlier. Meanwhile, prevention is probably the best way to minimize our chances to make the front page with a disgruntled employee affair. Stay up to date not only with patches, but with new technologies, new security models and new concepts. What's secure today will probably be insecure tomorrow.

References

2003 CSI/FBI Computer Crime And Security Survey
http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2003.pdf

Computer Viruses, trojan Horses and Logic Bombs
http://lrs.ed.uiuc.edu/wp/crime/viruses.htm

Information Security Glossary – Logic Bomb
http://www.yourwindow.to/information-security/gl_logicbomb.htm

Mcafee Virus Information - WM/Theatre.A
http://us.mcafee.com/virusInfo/default.asp?id=description&dtop=&virus_k=98311

Yahoo! Movies – Safe House
http://movies.yahoo.com/shop?d=hv&id=1802834627&cf=info&intl=us

Advanced NT Security Explorer
http://www.crackpassword.com/products/prs/otherms/ntsecurity/

LC4
http://www.atstake.com/research/lc/

SecureWave | Secure EXE FAQ
http://www.securewave.com/products/secureexe/faq-exe.html

CBS News | Logic Bomb Dropped On Brokerage
http://www.cbsnews.com/stories/2002/12/18/tech/main533450.shtml

disLEXia
http://md.hudora.de/blog/

disLEXia - Computer systms hit by logic bombs (1987-11-13)
http://md.hudora.de/blog/guids/26/13/200312301139518780144.html

disLEXia - UK Logic Bomb Case is ThrownOut (1988-01-12)
http://md.hudora.de/blog/guids/99/02/00312301322319529328.html

Salami Scam - InfoSecPedia
http://www.securitygroup.org/wiki/wiki.php?title=Salami_scam&PHPSESSID=a023b4f32d6aed419651bd3f62837036

Clearlybusiness

http://www.clearlybusiness.com/e_business/vs_trojan_horses_worms_&_logic_bombs.jsp

Australian Communication – Handbook 12 : Malicious Software
http://www.dsd.gov.au/_lib/pdf_doc/acsi33/HB12p.pdf