UNIVERSITY OF ZAGREB

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

SEMINAR FOR THE "COMPUTER FORENSICS" COURSE

[2017/2018]

The Morris Worm (1988)

Ana Brassard

Zagreb, January 2018

Table of Contents

1. Introduction

Longer than we would like to admit ago, “The Internet” had yet to be invented, home computers were a strange machine your uncle brought home from work one day, and the words “Why do you have all these toolbars installed?!” had never been bemoaned. Slowly, but surely, the computer started to show its potential in everyday applications, and optimistic dreamers predicted: “*The Computer will be anything from a home database with recipes and first aid instructions, to a personalized newspaper, babysitter, and even allow grocery shopping through the television. By the end of the 1990s, surely every single American household will have intelligent homes automatically brewing your morning coffee while you paid for your wife’s purchases electronically*[1]*”!* Come the late 1980s, however, many a geek’s heart had been broken. Only about 15% of American households had a computer, with most skeptical of their usefulness. Nevertheless, as we know, they were only wrong in time frame. During the time through which we saw tremendous advancement of this curious machine, the computer world went through many major transformative breakthroughs. One of these major points in advancement was the creation of networks which would later become what we now call the Internet. Computers became connected. More and more people joined the network, transforming the way we communicate and exchange information.

Around that time, sometime before November 2, 1988 to be exact, a Cornell University student named Robert Tappan Morris noticed some security flaws in the current computer network. He decided to demonstrate the effects of it by releasing a small piece of code to the Internet…

---

[1] Yes, really: "1999 A.D." (1967)

2. The Internet Worm of November 2, 1988

The Internet worm of November 2, 1988, or The Morris Worm, was one of the first computer worms distributed over the Internet. The worm is infamous for its scope and devastating effects it had on the computers connected to the Internet at the time. It is speculated that 6000 computers became infected, mostly owned by institutions such as MIT, NASA, and the Pentagon. This claim, however, was a statistic cooked up by estimating that there were around 60,000 computers connected to the Internet, and that around 10% of them were infected [1]. It is also speculated that the infection caused around $1,000,000 in damages, mainly in computer downtime during the 72 hours it took to purge the worm[2]. The actual number of infected computers remain unknown, but this did not stop the media from reporting the Worm as the most devastating and far reaching computer worm in the history of malware so far [2]. This was also a novelty – computer viruses rarely got any attention from the public due to computers not being as widespread at the time, as mentioned in the introduction.

Its author, Robert T. Morris, claims to have never had malicious intent, in the sense that the program was not designed to do any damage [3]. Morris meant to create a small program that would spread widely and use minimal computer resources, while remaining difficult to detect and remove. The implementation of these specifications contained an error which resulted in the program behaving erratically and consuming so much resources that the infected computers became unusable until the infection was purged.

[2] It is interesting to note that John McAfee, whose name rings familiar to most computer users today, claimed that cleaning up the network and fixing the systems flaws cost around $96 million [7]. Others further inflated the cost approximation to $186 million. These exaggerated estimates were likely publicity stunts designed to feed the rising fear of computer viruses – and in turn sell more anti-virus software.

2.1 The Spreading Mechanism

An interactive version of the map of the algorithm shown in Figure 1 can be found here.

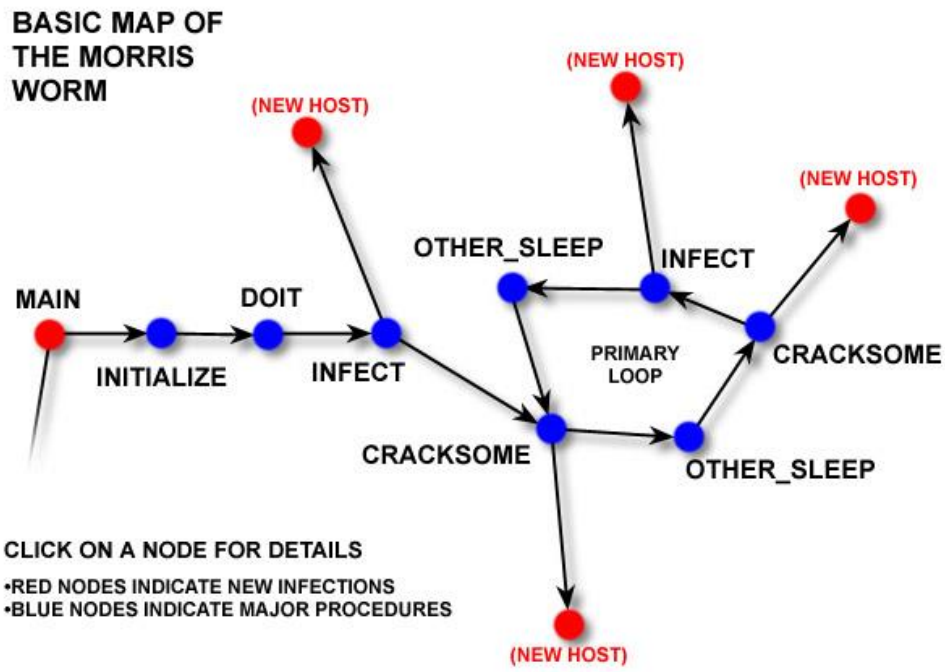

Figure 1. Map of the Morris Worm algorithm [4]

2.1.1 The Error

Morris was greatly concerned with making sure the worm remained undetected. At the beginning of the main function, the Worm goes through some steps that make sure that the running program is virtually invisible. In addition, he wanted to avoid infecting a computer more than once, so that the program kept its resource consumption to the minimum. To avoid re-infecting a computer where a copy of itself already existed, the Worm first checks if there already is a copy in the current computer. If this function returns "Yes", it moves on to the next connected computer instead of copying itself. However, Morris felt that infection might be prevented by creating a process that always returned "Yes" to this check, so he added a ***one in seven chance that the worm copies itself regardless of passing this check.***

Additionally, all subsequent copies of the worm ***skip this check entirely***. This is the part that caused computer failures.

2.1.2 The Infect Routine

The Worm tries to run a new instance of itself on the current machine through three attack attempts [4]. This is the part where the security flaws Morris noticed are exploited.

- **`try_rsh`** – Creates a duplicate process which attempts to remotely execute on the target machine. If successful, the target machine can now receive a new Worm process.
- **`try_fingerd`** – Exploits the UNIX `fingerd` function who reads an argument string, sends the argument to the target machine, and returns a response. This function is exploited by using the fact that it reads the argument using the C command `gets()` which has a 512-character buffer that is not protected against overwriting. The Worm calls `fingerd` with a 536-character argument, ***and the additional 24 characters end up being copied into the system stack***, which controls which functions to call next.
- **`try_sendmail`** – Takes advantage of a flaw in the program used to send email. It contained a `DEBUG` flag which was used while the program was being designed and never removed – it allowed sending an email to a process. The Worm sends an email with a recipient string that deletes the header of the message being sent ***and directly copies the body of the message,*** containing code to pull a copy of the Worm, ***into the system stack***.

2.1.3 The Cracksome Routine

Additionally, the Worm contains functions intended for breaking into user accounts which were weakly password protected. These functions are mostly guesswork, trying out common passwords for every account. It accesses the `/etc/passwd` file which contains encrypted passwords, tries encrypting a password guess using its own encryption algorithm, and compares the result. This way, the worm "tries out" many password attempts without triggering multiple failed login attempt warnings. It also goes through a small dictionary of words that might be good

candidates for passwords. To put things in perspective, with 1988 hardware, processing the entire dictionary containing 432 words is estimated to have taken ***at least 9 hours*** [4].

2.2 The Effects of Getting Infected

Had the Worm worked as intended, being infected with it would have been completely harmless. It was supposed to use minimal resources, and only spread itself through connected machines. It did not intend to harm any files on the machine, save or transmit guessed passwords, attempt to gain root access to the computer or leave copies of itself to be executed later (timebombs). It only attacked certain types of machines and did not cause any physical damage to them. [4]

However, the code did **not** work as intended.

As noted in a thorough analysis of the Worm shortly after it was dealt with, the code seemed to be incomplete, buggy, and contained some stylistic quirks that could indicate an inexperienced programmer, even going as far as saying that "the code seems to be the product of an inexperienced, rushed, or sloppy programmer" [5][3], perhaps in response to media at the time calling the author a "Computer Wizard" or "Genius" .

The crucial error in the code, as previously described, was that the Worm effectively reproduced itself infinitely within a same machine. This caused the Worm to use up all available resources on the machine and render the computer unusable. This

[3] The author of this sentiment mentions, in a footnote, that it was speculated that the motivation behind the program might have been to impress someone, who may as well be Jodie Foster. He expresses concern over a disturbing fact – that not everyone with access to computers is rational and sane, which future attacks may reflect. This was written in 1988. Another interesting comment is about the dominant assumption that the author was singular and male: "*Are we so unaccustomed to working together on programs that this is our natural inclination? Or is it that we find it hard to believe that more than one individual could have such poor judgement?*"

took only **90 minutes** from the time of infection [4]. This was especially problematic since it was computers in research and military institutions that were getting infected.

2.3 Solution and Prevention

Soon after deploying the Worm, Morris realized that he had greatly underestimated the rate of infection. He attempted to communicate a way to stop infection, but the Internet had already become too congested at that point and his message did not reach the intended destination [3]. The infection was eventually purged by partitioning the Internet for several days to prevent further infection. The fix used the same mechanisms of communication that enabled to Worm to spread, that is, it was done over the Internet [5]! Last and not least, the exploited security flaws have since then been patched, meaning that an identical attack today would have no effect.

2.4 Historic Significance

Robert Morris ***was tried and convicted of violating United States Code: Title 18, the Computer Fraud and Abuse Act*****,** and was sentenced to three years of probation, 400 hours of community service, a fine of $10,050, and the costs of his supervision. This was the first time an individual was tried under this Act [6]. Some disagree with his conviction, claiming that the code contained no commands that were intended to harm the target, and that he had forced some security flaws into getting fixed. Others point out that this could have been done some other way [6].

On the other hand, the incident inspired the formation of the Computer Emergency Response Team which had a crucial role in ensuring network and software security. They published a variety of announcements, analyses, fixes and warnings concerning software and networking vulnerabilities. There is a direct link between the Morris Worm and a major initiative that attempts to uphold security standards. These facts are key points of discussion about the Worm from an ethical standpoint, which could raise some interesting questions.

3. Conclusion

The Morris Worm incident teaches that code should always be thoroughly tested and debugged before deploying.

Joking aside, the Morris Worm is an interesting piece of malware in several aspects: the context in which it appeared, the code itself, the effects of it running amok, the consequences and the lessons that are still relevant today. Malicious intent is not a prerequisite for malware to appear – sometimes it is the product of a simple error. In cases like this, it may not be completely clear that the author of a program exhibiting malicious behavior is guilty of an attempting an attack. Such cases should be handled with utmost care and honesty to correctly understand what the purpose of a program was, and what the damage is. In Morris' case, it was concluded that, although it was not meant to damage the machines per se, it was clear that Morris was consciously exploiting vulnerabilities to knowingly access computers without authorization. This is certainly hard to disagree with.

4. Bibliography

[1] P. Graham, "The Submarine," [Online]. Available: http://www.paulgraham.com/submarine.html#f4n. [Accessed January 2018].

[2] S. Bortnik, "Five interesting facts about the Morris worm (for its 25th anniversary)," welivesecurity, 6 November 2013. [Online]. Available: https://www.welivesecurity.com/2013/11/06/five-interesting-facts-about-the-morris-worm-for-its-25th-anniversary/. [Accessed January 2018].

[3] *UNITED STATES of America, Appellee, v. Robert Tappan MORRIS, Defendant-Appellant.,* 1990.

[4] C. Schmidt and T. Darby, "The What, Why, and How of the 1988 Internet Worm," July 2001. [Online]. Available: https://snowplow.org/tom/worm/worm.html. [Accessed January 2018].

[5] E. H. Spafford, "The Internet Worm Program: An Analysis," Department of Computer Sciences, Purdue University, West Lafayette, IN, 1988.

[6] C. Kelty, "The Morris Worm," limn, [Online]. Available: https://limn.it/the-morris-worm/#edn1. [Accessed January 2018].

[7] "Morris Internet worm," Vmyths, 18 January 2001. [Online]. Available: http://www.vmyths.com/hmul/5/7/. [Accessed January 2018].