Technical review of the 7Devils – Cyber Security Case Study – Robert Morris Internet Worm

2023-09-19T01:58:41Z

Robert Tappan Morris[1] released the first *worm*[2] on the Internet on 2nd November 1988 at 8:30 PM. At that time, when only around 60,000 computers were connected to the nascent *Internet*[3] across 17 countries, the malware managed to infiltrate approximately 10 %, infecting around 6.000 computers. Many of the affected systems were rendered unusable or even shut down. Colleges, research centers, hospitals and the military were mainly affected. This event marked a significant milestone in the early history of cybersecurity. This technical review seeks to address four fundamental questions arising from this historic incident:

1. How did the Morris worm work?
2. What was the *payload*[4]?
3. How was the program identified?
4. How did security experts stop the Morris worm?

The Morris Worm employed a range of techniques to facilitate its self-propagation.

One aspect exploited a backdoor vulnerability within *Sendmail*[5], the Internet's electronic mail system. *Sendmail* was a program capable of running in various modes, one of them being a background daemon[6]. It enabled mail delivery directly to the daemon rather than to mailbox files, a functionality often utilized for tasks such as configuring automated vacation responses. Accidentally, a misfeature was introduced: if *Sendmail* was compiled with *DEBUG* flag on and if the sender of a mail asks the daemon to go in debug mode, then the *Sendmail* allowed the sender to send a sequence of commands instead of a recipient's address.

Another aspect of his code exploited a *Unix* utility called *finger*[7], to identify network users. This utility was used to obtain general user information. Running as a background daemon like *Sendmail*, the worm exploited finger by overrunning the buffer the process used as its input.

A third aspect was the use of remote command interpreters over the network, *rsh*[8] and *rexec*[9]. The worm exploited the fact that there is a high probability that a password for a local user for an account on a remote machine will be the same as its local password.

A fourth aspect of the worm was the exploit of publicly readable files that stored encrypted user passwords. It would allow the worm to rapidly try to guess passwords using a *dictionary*

[1] https://en.wikipedia.org/wiki/Robert_Tappan_Morris
[2] Self-replicating computer program.
[3] https://en.wikipedia.org/wiki/Internet
[4] https://en.wikipedia.org/wiki/Payload_(computing)#Security
[5] https://en.wikipedia.org/wiki/Sendmail
[6] https://en.wikipedia.org/wiki/Daemon_(computing)
[7] https://www.ibm.com/docs/en/aix/7.2?topic=f-finger-command
[8] https://en.wikipedia.org/wiki/Remote_Shell
[9] https://www.ibm.com/docs/vi/aix/7.2?topic=r-rexec-command

Martin Nizon-Deladoeuille, mma18@hi.is (exchange student – University of Iceland).

attack[10]. After discovering a user's password, it could then attempt to gain access to other computers in the network, on which the user had the same account with the same credentials.

Written in the *C coding language*[11], the payload of the Morris worm consisted of several components designed to enable its self-replication and propagation across computer networks. Once the worm infected a host machine, it attempted to locate other potential hosts to infect and then exploit vulnerabilities (notably those seen in the preceding paragraph) to spread further.

Morris aimed to prevent multiple infections on a single computer, which could lead to performance degradation. He devised a mechanism where, in the event of two worm copies coexisting on a device, a virtual coin toss would determine which copy persisted and which self-destructed. Unfortunately, due to a programming error, this safeguard failed upon the worm's release, resulting in extensive damage. Multiple systems had multiple versions of the worm running on them, causing them to slow down or even crash.

The Morris worm was identified through a combination of unusual network behavior and reports from system administrators about their systems behaving erratically. As it rapidly spread through the early internet, it caused significant congestion and slowdowns. Network anomalies, increased system loads, and the worm's attempts to exploit vulnerabilities drew the attention of cybersecurity experts and system administrators. Subsequent analysis of its code and behavior led to its identification and the discovery of its propagation methods.

Many researchers at Berkeley[12], MIT[13] and Purdue[14] studied the worm and uncovered how it works and released some fixes and patches within a day. The way researchers fixed the Morris worm using security patches has become a standard way to deal with the security vulnerabilities exploited by hackers to this day. It is imperative to keep systems and software up to date to help avoid getting hacked.

The Morris worm marked the end of the age of innocence for the Internet. Up until that point, there were many public servers that were easily accessible for sharing research and other information. After the Morris worm, the community's collective eyes were opened to the potential damage a virus, or a worm could cause.

Curated Information

1. The publication 7Devils – Cyber Security Case Study – the Robert Morris Internet Worm from Dr. Leibrock gives the key points about the story of the first worm.
2. A study on the Morris Worm from Akshay Jajoo, published on 15 December 2021 gives many details about loopholes and misfeatures, as well as the operation of the Morris worm.
3. The video The FIRST Computer Worm: The Morris Worm (1988) – WhatTheHack gives an overview of the Morris worm.

[10] Brute-forcing using a list of common list of common passwords.
[11] https://en.wikipedia.org/wiki/C_(programming_language)
[12] https://www.berkeley.edu/
[13] https://www.mit.edu/
[14] https://www.purdue.edu/