Demonstration of Operating System Security

Ryan Scott

Grand Canyon University  
ITT-305 Information Security I

Prof. Scott Knapper

# Abstract

This paper aims to describe how unauthorized access to the resources of operating systems can be accomplished through the use of techniques such as buffer overflow exploits and rootkits. It will also explore some of the security measures that are in place on Windows, Linux, and Mac OS X operating systems in order to prevent such breaches.

**Buffer Overflow**

* **Stack**
  + Computer science heavily relies on stacks
  + Many buffer overflows occur in this area
  + One vulnerability is that they require a finite amount of buffer space to be allocated, (Computerphile, 2016)
* **Return address**
  + An address is read in and points to a location in the code that will run it.
  + Bypassing the buffer into the return address allows you to overwrite it with your own address, (Computerphile, 2016)
  + There could be a serious issue if there is a virus located at this address
* **Allocated buffer space**
  + A finite amount of space must be allocated for the buffer on the stack
  + In the event that we write something into the buffer that is longer than the buffer, we will write into new information we are not supposed to be able to reach, such as the return address, (Computerphile, 2016)
* **Segment fault**
  + You will receive this message when you try to access something in memory that you shouldn't, (Computerphile, 2016)
  + Linux right of passage
* **GDB**
  + Linux command line debugger that provides very useful information
  + Shows a lot of low-level assembly information
  + Can disassemble information into machine code that would be running to the CPU, (Computerphile, 2016)
* **No-op sled** 
  + A gigantic “slide” for your OS to fall into.
  + Machine instruction for “just move to the next one”
  + The malicious code will tick along wherever we land in the No-op sled, (Computerphile, 2016)

**Code Red, SQL Slammer, & Buffer Overflow Exploration**

**1a. Result and cost of the attack of Code Red?**

* 1. More than 359,000 computers that were connected to the Internet were infected in less than 14 hours as a result of the Code-Red worm, (Moore et al., 2002).
  2. There are an estimated 2 billion dollars spent globally on containing and controlling the Code-Red (CRv2) worm that spread rapidly throughout the Internet on July 19, 2001, (Moore et al., 2002).

**1b. Result and cost of the attack of SQL Slammer?**

1. The simplicity of the SQL Slammer’s infection method combined with the speed at which potential computers were probed allowed SQL Slammer to double in size every 8 seconds, (Gregory Travis et al., n.d.).
2. It is estimated that Slammer will cost approximately one billion dollars in constant dollars, an order of magnitude more costly than the Morris worm it is similar to,(Gregory Travis et al., n.d.).
3. **What are the most common languages used for writing buffer exploits and why?**
   1. Because C and C++ lack built-in safeguards against overwriting and accessing their memory, they are highly vulnerable to buffer overflow attacks, (Imperva, 2019).
   2. Safety mechanisms are built into languages such as PERL, Java, JavaScript, and C# to prevent buffer overflows, (Imperva, 2019).
4. **How have operating systems been improved to prevent intentional buffer overflows?**
   1. By randomizing address spaces, buffer overflow attacks are unable to determine the location of executable code in address spaces (Imperva, 2019).
   2. By marking memory regions executable or non-executable, data execution prevention will prevent running code in non-executable regions, (Imperva, 2019). The stack itself can be an example of one such region (Computerphile, 2016).
   3. The structured exception handler overwrite protection (SEHOP) provides protection against malicious code attacking the hardware and software exception management systems. It prevents an attacker from exploiting the SEH overwrite technique, which overwrites the exception registration records stored on a thread's stack with a stack buffer overflow, (Imperva, 2019).

**Rootkit Demonstration**

1. **Why was Windows XP used as the target machine in the video?**
   1. Security vulnerabilities that allow the demonstration via brute force, (Kawa, 2012).
   2. The XP machine will have open ports that we will try to exploit using an nmap scan, (Kawa, 2012).
2. **How does the Beast program work and how does it work?**
   1. The Beast acts as a gateway for other network nodes on the Windows XP machine, (Kawa, 2012).
   2. Beast provides the hacker with access to all files on other nodes, including passwords, documents, and webcams, (Kawa, 2012).
3. **Describe how certain command options in FU make it useful as a rootkit.**
   1. FU is a kernel-mode rootkit, allowing kernel data structures to be modified, (Kawa, 2012).
   2. This is immensely useful for a malicious piece of code that is interested in obscuring itself within the process list, such as the Beast program, (Kawa, 2012).
4. **How is the Black Light security tool shown in the video different than the normal Process List utility of Windows?**
   1. In the task manager, the Beast rootkit was hidden in the list of processes, (Kawa, 2012).
   2. With the help of Black Light, hidden processes can be detected, (Kawa, 2012).
   3. When the problem has been identified, it can be cleaned and taken care of quickly and easily, (Kawa, 2012).

# **References**

Computerphile. (2016, March 2). Running a Buffer Overflow Attack - Computerphile. YouTube. Retrieved November 7, 2022, from <https://youtu.be/1S0aBV-Waeo>

Gregory Travis, Ed Balas, David Ripley, & Steven Wallace. (n.d.). Analysis of the “SQL Slammer” worm and its effects on Indiana University and related institutions. Advanced Network Management Lab Cybersecurity Initiative Indiana University. Retrieved November 7, 2022, from <https://uh.edu/tech/cisre/resources/ia-resources/_files/7033/Week10/slammer.pdf>

Kawa, S. (2012, May 31). Rootkits Demonstration. YouTube. Retrieved November 7, 2022, from <https://www.youtube.com/watch?v=6CFnK20EyUU&feature=youtu.be>

Moore, D., Shannon, C., &amp; Brown, J. (2002, November 1). Code-red: A case study on the spread and victims of an internet worm. CAIDA Resource Catalog. Retrieved November 7, 2022, from <https://catalog.caida.org/paper/2002_codered>

What is a Buffer Overflow | Attack Types and Prevention Methods | Imperva. (2019, December 29). Learning Center. Retrieved November 7, 2022, from <https://www.imperva.com/learn/application-security/buffer-overflow/>