Buffer Overflow in SSH

Attack of the Week

Source: CERT Advisory CA-2002-36

Description

Multiple SSH2 servers and clients do not properly handle large packets or large fields, which may allow remote attackers to cause a denial of service or possibly execute arbitrary code via buffer overflow attacks.
What is a Buffer Overflow?

- Occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold.
- The extra data can overflow into adjacent buffers.
- The extra data may contain codes designed to trigger specific actions.

Buffer Overflow Exploit

- C function arguments and return address are written to stack.
- In a buffer overflow exploit, code is written beyond the return address and function call arguments.
- The return address is modified to point to the beginning of the code.
- When the function call returns, the attacker's code (e.g. a shell) is executed.
- Difficult to program, easy to exploit.
Impact of Attack

- SSH is a secure low-level transport protocol
- SSH provides strong encryption, cryptographic host authentication, and integrity protection
- SSH is used for
  - Secure remote login
  - “Tunneling” of insecure TCP/UDP traffic
- Many programs rely on SSH for secure transport
- SSH is critical for many applications
- Compromised SSH affects a variety of applications/services

Countermeasures

- Apply patch to vulnerable software
- Prevention: Good programming practices
  - Avoid/use carefully C subroutines without boundary checks: strcat(), strcpy(), sprintf(), vsprintf(), bcopy(), gets(), scanf()
- Boundary checks
- Non-executable stacks
Sources/Further Reading

- http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-1359
- http://www.phrack.org
- http://www.cultdeadcow.com/cDc_files/cDc-351/