Enriching security toolbox in Solaris with Netcat

Vladimír Kotal
Revenue Product Engineer (Solaris Security)
Sun Microsystems Inc.
How it all began?

- **CVE-2006-4343**
  - NULL pointer dereference in OpenSSL
  - Need to reproduce and test the fix
  - Exploit was provided
    - `perl -e 'print "\\x80\\xec...", "\\x00"x"5","A"x"512"' | nc -lp 443`
  - Now what?
Which one to choose?

- Many **Netcat** implementations
  > nc(1) is merely a genre than a program
- **OpenBSD's nc** won
  > compared 4 most commonly used implementations
    - criteria: coding level (cleanliness, style, robustness), features, license, maintenance history
Where to put it?

- Solaris is made of *consolidations*
  - ON (OS-Net) aka kernel+libraries, SFW (Apache, Samba, ...), Install, etc.
- OpenSolaris ON gate was chosen
  - nc(1) is small enough
  - development cycle is over
    - since like 1997 or so
  - future changes will make it more tightly integrated with Solaris
- Where it lives?
  - \$SRC/cmd/cmd-inet/usr.bin/nc/
Input scrubbing

• Code review

• **Architectural Review**
  > determine what is interface, assign stability level (according to Interface taxonomy) to it

• OpenSource Review
  > performed by lawyers with data supplied by engineers (license check)

• Testing
  > set of unit functional per option tests
    - executed by hand
Code review (peer review)

- Correctness of code
- Secure programming techniques
- Tools
  - C-style check via `$SRC/tools/scripts/cstyle.pl`
    - guards official style (`cstyle.ms.pdf`)
  - `$SRC/tools/scripts/webrev.sh`
    - poor man's source changes comparison
ARC review

• Netcat integration
  > covered by PSARC 2007/389
  > most commonly used options are Committed

• Prerequisite: err/warn in libc
  > “err.c does not belong here. Why don't you add it to libc?”
  > PSARC 2006/662
  > [v]err[x](), [v]warn[x]() function family
Our modifications so far

- Strip BSD specific features
  - TCP_MD5SIG, jumbogram support, arc4*(), SO_REUSEPORT, {read->get}passphrase()

- Little bugfixes
  - Better and more verbose messages
  - Better usage corner case handling
  - Be good IPv6 app
    - listen on both AF_INET[6] wildcard sockets by default

- Man page tweaks
  - RBAC integration, SMF coverage (inetd(1M) is a set of services), more precise usage spec (stems from PSARC case)
Testing

• **Bryan Cantrill** in Developing Solaris:
  > “Have you tested your change in every way you know of and how? If not, do not go any further with the integration unless you do so.” (rephrased)

• **Unit tests**
  > cumbersome when performed by hand
  > **Test suite** needed
    - CTI-TET used as a framework
    - basic functionality tests (data transfer)
    - each option has a test case with several test purposes (some of them performing negative tests)
What's in the works

- I/O enhancements
  - buffer size control, more flexible EOF event handling
  - PSARC fast-track case is coming soonish
- Test suite review
  - prototype ready
  - to be integrated into ontest-stc2 and opensourced
Future of nc(1) in OpenSolaris

- **Protocol extensions**
  - IPsec (persock, bypass ?), SCTP
    - SSL not needed, openssl(1) handles basic cases just fine
- **Execute external program (-e)**
  - Yes, the dreaded GAPING_SECURITY_HOLE #define (in original nc110 implementation)
    - Instant backdoor? “Pure bunkum” to quote anonymous senior ON developer
- **Traffic redirector (?)**
  - read_write() is almost ready for it
Come to hack it too!

- Once in OpenSolaris it is open to everyone
- May seem like a niche but it's not
  - normal users aside, `nc(1)` is used by test suites, other system components (`libvirt uses nc` for remote hypervisor access)
  - programs like `nc` are great learning ground
  - Proof that anyone can find a place in OSol to work
Got some incoming data, er, questions?

Vladimir Kotal

http://blogs.sun.com/vlad/