Netmap2 Talk / Project

by Javantea aka. Joel R. Voss
Feb 23, 2006

Start of the talk: try to get people motivated with a good talk on protocol hacking. Explain Netmap2's uses, show a use of it. Explain the format of popular services: DNS, SSH, HTTP, BitTorrent, UDP Session Development. Explain the problem of writing services. Explain the Neg9 CTF project. We're done, split into groups.

2 - 4 teams Each team needs at least one coder/developer/guru. Each team needs 1-3 designers who do little/no code. One person can be the designated group teacher if they want or they can teach on demand / as necessary.

Each team must design and build a protocol in 1 hour. The protocol must input user data, modify it, and return data.

The code must be original, written tonight, open source, and easily portable to Linux ("the server"). Docked points for problems on other computers.
When the client and server are written, you must put it up on "the server".
You then must publish the source of client and server as well as design docs. Each team downloads them, and then attacks the servers with their evil clients. The teams have 1 hour to attack and publish exploits for their opponent's servers.

Criteria for judging servers:

- Is your service authenticated and/or public?
  - (10) Authenticated
  - (5) Public
- How much data are you giving to unauthenticated/public users?
  - (1) 1kB
  - (5) 1MB
  - (10) 1GB
- How much data are you giving to authenticated users?
  - (1) 1kB
  - (5) 1MB
  - (10) 1GB
- How useful is the data?
  - (1) Static
  - (5) Dynamic
  - (10) Terribly awesome
- How resilient is your protocol to bad data?
  - (1) Seems to work.
  - (5) Works well so far.
  - (10) No possible bad data attack.
- How extensible is the server?
  - (1) A little extensibility.
  - (5) Modules quite simple.
  - (5) Everything is a module.
- How backwards compatible are the extensions?
  - (-2) Clients crash after extension add.
  - (2) Owch, client rewrite time.
  - (5) Clients work after add.
  - (10) Client work after major extension change.
- How many minutes does it take a person to read INSTALL and configure it?
  - (5) <15 seconds
  - (2) <2 minutes
  - (2) <10 minutes
  - (-2) <30 minutes

Slogans for the whiteboard:
Sally might be your best friend, but don't trust her input data.
Measure twice, cut once.
Mind your Ps and Qs.
If it looks like a duck and quacks like a duck, it might be a duck, but it might also be a trojan duck.

Criteria for judging clients:

- How quickly does it run?
  - (-1) User limited
  - (2) Instantly
  - (2) Slowly
  - (5) Connection limited
- Where does the data go?
  - (2) Standard out
  - (2) File
  - (5) GUI
- How does the client know what they're getting?
  - (2) The server knows best
  - (5) Command line args
  - (2) Other
- Does the client trojan another program?
  - (5) Yes
  - (3) No
- Is the client a trojan?
  - (-10) Yes
  - (5) No
- Is the client trojaned?
  - (-10) Yes
  - (5) No
- Can the client send shellcode?
  - (5) Yes
  - (3) No

Criteria for judging exploits:

- Does the exploit do something?
  - (5) Change data on the server.
- (10) Crash the server.
- (15) Own the server completely.
- Does the exploit deny service?
  - (2) At least one user.
  - (5) A lot of users.
  - (10) Undetectable/unstoppable.
  - (-2) < 1kbps for 5 seconds
  - (-2) < 10kbps for 5 seconds
- Is the exploit a trojan?
  - (5) Yes.
  - (10) Undetectable.
  - (25) Automatic breach.
- Is the exploit a worm?
  - (25) Yes. (Please don't run it.)
- Does the exploit have a payload?
  - (15) Yes.

Prizes:

First Prize: Hacker Book from Univ. Washington Bookstore.
Close Second: $15 gift certificate to ThinkGeek.
Close Third: A paid copy of the open source distro of your choice (<$40).
Obsfucated constants prize: UNI Washington $5 calling card.
Unextensibility prize: Slightly used crayola crayons.
Incompatibility prize: Ubuntu CD.

Two rules exist for the prizes:
1) Allow someone temporary use of the prize upon reasonable request.
2) Any prize not won (close second/third, or special) will become a prize for next meeting.