acmqueue

The Network Protocol Battle

A tale of hubris and zealotry

Dear KV,

I've been working on a personal project that involves creating a new network protocol. Out of curiosity, I tried to find out what would be involved in getting an official protocol number assigned for my project and discovered that it could take a year and could mean a lot of back and forth with the powers that be at the IETF (Internet Engineering Task Force). I knew this wouldn't be as simple as clicking something on a Web page, but a year seems excessive, and really it's not a major part of the work, so it seems like this would mainly be a distraction. For now, I just took a random protocol number that I know doesn't conflict with anything on my network—such as UDP or TCP—and things seem to work fine. I guess my real question is why would anyone bother to go to the IETF to ask for this unless they were a company that could waste someone's time on an e-mail campaign to get a properly assigned number?

Waiting

Dear Waiting,

Let me begin by complimenting you on the fact that you actually went so far as to find out how one might do this correctly. (I am sure that many readers have just spit coffee on their screens, because none of them can remember the last time I complimented a writer to this column.) I compliment you because just recently I came across someone who knew the right thing to do, and then did exactly the opposite.

Because of some of the assumptions present in the original design of the Internet, some parts of the IPv4 packet header are far more precious than others, and, while the limitations of the 32-bit network address get the largest amount of attention, the 8-bit protocol field is equally as important, if not more so. With an 8-bit field, we can layer only 255 possible protocols on top of IPv4, which may seem like a lot, and since most people assume that all IP packets carry only TCP, protocol 6, there is plenty of space. It turns out that more than half of the numbers have been used for one protocol or another, leaving only 109 for use by authors of new protocols. Another problem is that IPv6, the nominal savior of the Internet, with its wider network addresses, still uses an 8-bit protocol field, so we're not getting any more space any time soon.

The protocol field can be seen as a commons for the Internet, so let me tell you a tragedy, and one that didn't have to happen. It is a story of hubris and zealotry, and unsurprisingly, involves the collision between corporations and open source.

Sometime in the late 1990s, a group of companies got together and proposed a protocol that would be standardized within the aegis of the IETF. It's not particularly important what the protocol does, but it's called VRRP (Virtual Router Redundancy Protocol) and exists so that two or more routers can act as peers in a fail-over scenario. If one router fails, another router discovers this via a means described in the protocol and takes over for the failing router. After the standard was published, two companies—Cisco and IBM—both claimed to have patents to some of what

the protocol did. Cisco released its claimed intellectual property under a RAND (reasonable and nondiscriminatory) license. In nonlegal terms this means that people could implement VRRP, and Cisco would not chase them down with expensive claims. RAND licenses are often used in software standardization processes.

Unfortunately, there is a segment of the open source community that is incapable of playing well with others, when those others don't play the way they want them to. For those who have not had to deal with these people, it's a bit like talking to a four-year-old. When you explain checkers to your niece, she might decide that she doesn't like your rules and follows her own rules. You humor her, she's being creative, and this is amusing in a four-year-old. If you were playing chess with a colleague who suddenly told you that the king could move one, two, or three places in one go, you would be pissed off, because this person would obviously be screwing with you, or insane.

Have I lost my mind?! What does this have to do with VRRP or network protocols?

The OpenBSD team, led as always by their Glorious Leader (their words, not mine), decided that a RAND license just wasn't free enough for them. They wrote their own protocol, which was completely incompatible with VRRP. Well, you say, that's not so bad; that's competition, and we all know that competition is good and brings better products, and it's the glorious triumph of Capitalism. But there is one last little nit to this story. The new protocol dubbed CARP (Common Address Redundancy Protocol) uses the exact same IP number as VRRP (112). Most people, and KV includes himself in this group, think this was a jerk move. "Why would they do this?" I hear you cry. Well, it turns out that they believe themselves to be in a war with the enemies of open source, as well as with those opposed to motherhood and apple pie. Stomping on the same protocol number was, in their minds, a strike against their enemies and all for the good. Of course, it makes operating devices with both protocols in the same network difficult, and it makes debugging the software that implements the protocol nearly impossible.

In the end the same thing is going to happen as happens when your four-year-old niece up-ends the checkers game in frustration. She runs away crying, and you're left to pick up the pieces. A few of us now have to take this protocol and actually get it a proper protocol number and then deal with the fact that legacy devices are still using the old, incompatible protocol.

Now I think you see why I wanted to compliment you. Doing the right thing in the commons is good for all of us. Thanks for not being a jerk.

KV

Dear KV,

One of my coworkers seems to be completely allergic to useful tools. For example, rather than extend a program to interpret new data properly in one of our log generators, he instead memorizes the format and reads it off the screen when something breaks. He seems to take great pride in this, but I'm not really sure why. Wouldn't a tool make more sense?

Hexed by Dumps

Dear Hexed,

I'm not really sure how many variations on the theme of "Tools are Useful" I will eventually write, but clearly my message has not penetrated to the depth that I would like. In part, I think the problem is that geeks—and I include myself in this group—get off on feeling that they know

things—in particular, things that they think others don’t know. You can see this in the in-jokes we tell one another, such as the number of times the number 42 appears in code. It is this love of arcana, I think, that leads people to be proud of the fact that they can get along by reading hex dumps.

Of course, the problem is that some ways of looking at data are more prone to error than others. For example, as someone who looks at a lot of network packets, I can tell you that the following

```
0x0000:  4500 0045 bf49 0000 ff11 7902 c0a8 010a
0x0010:  c0a8 0101 d688 0035 0031 8a10 39e6 0100
0x0020:  0001 0000 0000 0000 0c70 3034 2d75 6269
0x0030:  7175 6974 7906 6963 6c6f 7564 0363 6f6d
0x0040:  0000 0100 01
```

is an IPv4 packet, and I can then scan through it to find the IP addresses and other bits that I need to know to diagnose what’s wrong with a host’s network connection. Compared with the above, though, I would rather read this:

```
14:19:02.997326 IP (tos 0x0, ttl 255, id 48969, offset 0, flags
[none], proto UDP (17), length 69)
192.168.1.10.54920 > clearspot.domain:
[udp sum ok] 14822+ A? p04-ubiquity.icloud.com. (41)
```

Sure, the second example is also gobbledygook to the average person, but it’s far easier to read and understand for the professional who is trying to get a job done. It’s also just as compact as the first example, so an argument cannot even be made that the hex dump is quicker to read. The real issue is that getting a job done is what so many geeks lose sight of when they’re preening themselves over their l337 hex-dump reading skills; it’s not about knowing the arcana, it’s about solving a problem. If you want to learn and work purely in arcana, then you should study the Kabbalah, or go on *Jeopardy*, where you can now be beaten by a computer. If you want to get a job done, then use or build the tool that will give you the clearest view of the problem.

KV

KODE VICIOUS, known to mere mortals as George V. Neville-Neil, works on networking and operating system code for fun and profit. He also teaches courses on various subjects related to programming. His areas of interest are code spelunking, operating systems, and rewriting your bad code (OK, maybe not that last one). He earned his bachelor’s degree in computer science at Northeastern University in Boston, Massachusetts, and is a member of ACM, the Usenix Association, and IEEE. He is an avid bicyclist and traveler who currently lives in New York City.

© 2011 ACM 1542-7730/11/1200 \$10.00