AT WHAT POINT DO OUR CYBERWAR TOYS BECOME WMD?

The other day, Ellen Nakashima reported on new cyberwar acquisition guidelines that will allow DOD, under certain circumstances, to deploy targeted exploits without the regular testing or oversight process.

> The rapid process will take advantage of existing or nearly completed hardware and software developed by industry and government laboratories. This approach could take several months in some cases, or a few days in others.
>
> [snip]
>
> Under the rapid plan, weapons can be financed through the use of operational funds, in "days to months," and some steps that ordinarily would be required would be eliminated. These include some planning documents and test activities, according to the report.
>
> The weapons may be designed for a single use or for some other limited deployment, and they would be used in offensive cyber operations or to protect individual computer systems against specific threats, said the report.

As she describes it, this rapid development will (is supposed to?) only be used in fairly targeted cases.

But what are the chances the speed and limited oversight lead to mistakes? What are the chances that our rush to roll out exploits leads us to set off some unintended consequences?

Consider Richard Clarke's explanation for how StuxNet escaped the narrow confines of the Natanz centrifuge facility it targeted.

> “It got loose because there was a mistake,” [Clarke] says. “It’s clear to me that lawyers went over it and gave it what’s called, in the IT business, a TTL.”
>
> “What’s that?”
>
> “If you saw *Blade Runner* [in which artificial intelligence androids were given a limited life span—a “time to die”], it’s a ‘Time to Live.’” Do the job, commit suicide and disappear. No more damage, collateral or otherwise.
>
> “So there was a TTL built into Stuxnet,” he says [to avoid violating international law against collateral damage, say to the Iranian electrical grid]. And somehow it didn’t work.”
>
> “Why wouldn’t it have worked?”
>
> “TTL operates off of a date on your computer. Well, if you are in China or Iran or someplace where you’re running bootleg software that you haven’t paid for, your date on your computer might be 1998 or something because otherwise the bootleg 30-day trial TTL software would expire.
>
> “So that’s one theory,” Clarke continues. “But in any event, you’re right, it got out. And it ran around the world and infected lots of things but didn’t do any damage, because every time it woke up in a computer it asked itself those four questions. Unless you were running uranium nuclear centrifuges, it wasn’t going to hurt you.”
>
> “So it’s not a threat anymore?”
>
> “But you now have it, and if you’re a computer whiz you can take it apart and you can say, ‘Oh, let’s change this over here, let’s change that over there.’ Now I’ve got a really sophisticated weapon.

> [first brackets mine, all others original]

Here's a cyberweapon presumably developed under the existing "deliberate" process, with full testing and oversight. If Clarke's description of the problem is correct, it's not so much a testing problem as an inadequate understanding of the environment—a failure to account for all those computers on which, because their clocks were not set properly, the TTL orders malfunctioned. And while StuxNet itself may not have done collateral damage, who knows what hackers who have gotten the code did with it?

So while StuxNet, with the benefit of time and testing, didn't do excessive damage when DOD's plans proved to be inadequate, who's to say that an exploit deployed with far less time—purchased for use—won't do more damage?

Also, note how much more quickly DOD appears to be moving to make sure it has lots of cyberweapons to deploy than it has moved to make sure it has the most rudimentary defenses against exploitation. Probably, when our cyberwar toys turn into a WMD, they'll hurt people in the Middle East or China. But given our rush into offensive cyberwar before we've protected ourselves, who knows?