A Constructive Build-the-Flag Contest

Matt Bishop

Summary. We have many people who know how to compromise existing systems, and capture-the-flag contests are increasing this number. We have a great shortage of people who know how to design and build secure systems. A contest to build secure systems to meet specific goals – a "make-the-flag competition" — could help with this problem.

The non-security of existing systems is widely known. In computer security curricula and competitions, a common exercise is to have students find flaws in existing systems. In some cases, the organizers of competitions make their own systems (such as DefCon's Clemency system). The goal of these exercises and competitions (called "Capture-the-Flag" or "CTF" contests here) is to teach students how easily vulnerabilities can be exploited, by having them do the exploitation; or to demonstrate their skills in doing so.

A variant of these CTF competitions is to provide the contestants with an existing system that is known to have vulnerabilities. They are given some period of time, such as a month, to harden the system so that any vulnerabilities cannot be exploited, and all attempts to do so are recorded. The systems are then attacked by other teams or a "red team" and the contestants are given points for the attacks they have blocked. These "Protect-the-Flag" ("PTF") competitions are more constructive than the CTF ones because the emphasis is on securing a system, not breaching it.

Consider the ultimate goal of security. It is to create systems that satisfy a specific set of requirements. The CTF competition focuses on showing an existing system fails to do this. A PTF competition focuses on protecting an existing but fundamentally non-secure system to prevent it from violating a set of security requirements. But neither of these do what a "secure system" is to do: demonstrate to some desired level of assurance that a system meets a set of specific requirements, including security requirements.

This suggests an alternate competition. Why not have the contestants design and implement a system to meet specific requirements, including security requirements? This competition, a "Make-the-Flag" (MTF) competition, has the contestant teams work from the

ground up to design and build a secure system, rather than work from the top down to take a system apart. Such a competition would of necessity involve a special-purpose system because designing and implementing a general-purpose system from scratch would take too long. Participants would be contestants or competitors who design and implement the systems; evaluators who score the system; judges, who score the contest; and the competition managers, who design the competition and manage it.

Of most importance to such a competition is the degree of specifications given. In all cases, the competitors must be told the requirements to be met. But there are two primary issues from the point of view of the contest developers.

The competitors may simply be told that their system must meet the given requirements, leaving how they do that completely up to them. In this case, the competitors must document their system well enough so the evaluators, who have never seen it, can verify that the system meet the requirements. The advantage to this approach is it offers the contestants the maximum degree of freedom, while teaching them to document their interfaces and other external features of their system thoroughly enough for the evaluators to be able to use their system. The disadvantage is that each system will likely have a unique interface, which will create more work for the evaluators.

The second is to include a specification of the interface as part of the requirements. This constrains the competitors in how the system is used, but it is realistic in that output requirements are common. Further, it eases the burden on the evaluators because they will not have to learn a new interface for each system.

A third way is to specify the hardware as well as the interface and other requirements. This is appropriate if the goal of the contest requires special purpose hardware for an interface. The contest can specify some or all of the hardware to be used.

These constraints are the only limits to the imagination of the people running the contest.

The problem we face now is not that we lack people who know how to attack systems. Indeed, part of our problem is that we have too many of them! An MTF competition shifts the focus to creating secure systems, and we lack people who can do that. It also forces students to pull together everything they have learned in computer science classes — software engineering, robust programming, networking, security, and so forth — to build a system that will be tested thoroughly for vulnerabilities. It will also encourage academia to put more emphasis on teaching this art of construction.

With a suitable reward system for the competition, and if as well done as CTF competitions, this contest could increase the number of people who can build secure systems.