Jesse Riddle

CCDC Write up

2013-02-18

My experience at the 2013 Collegiate Cyber Defense Competition

Security of information systems is an ongoing struggle, where both the sophistication of the attacks, and the stakes of an information breach are continually raised. While we successfully planned and implemented a strategy that seemed sure to bring us victory, there were a couple of critical flaws in our strategy at the Collegiate Cyber Defense Competition (CCDC). Teams competing at future CCDC events should note these flaws. I will first cover our strategy, and then visit on two major flaws in our implementation.

Our strategy, although not very complex, did require a background in configuring and updating common software services. Our focus was to make sure that known flaws in the services running on the systems, both those unintentionally included as exploitable bugs in common software, and those included intentionally by the red team, would not be our Achilles’ heel. We focused primarily on properly configuring software services, getting these services to work together, and both securing and updating these software services.

One by one, as we secured the software services, and removed suspicious scheduled tasks, as well as nonessential software from the virtual machines (“pods”). As we secured the services, we brought them online and exposed them to both the scoring engine, and attack from the red team. This gave us limited success during the entire competition. Our biggest point loss was early in the competition, and our biggest point gain was during the middle of the competition. Our strategy suffered from other critical flaws toward the end of the competition.

Our first critical flaw was neglecting the pre-existing installed user base. The systems were configured to authenticate against a central server, which contained several dozen users, protected by passwords that did not meet our password policy. While we changed the common passwords to access administrative interfaces throughout the network, more or less, we simply neglected this existing user base throughout the competition.

Our services may have been solid and secure, and were never taken down, but our users’ passwords had common dictionary words, and had been changed by the red team after having been cracked. This caused some visible “flapping” on the scoring engine, as it attempted to variably log in as one of the non-administrative users in our user base. We incorrectly attributed to difficulties with the scoring system that were happening during the competition.

 Attributing the inconsistency of the status of our services to problems with the scoring engine, as we could certainly reach them ourselves and see that they were operating. I, for one, began to question that the red team was even present in the competition. I was blind to their earliest attacks because of my own mistake.

My second task (after updating the software at the network gateway) was to configure an intrusion detection system (IDS) called “snort”. I had done so with Ruben’s help. I also configured it to log to a central server, internal to our network, so possible intrusions could be viewed there. The problem was that, although I had updated and configured the gateway successfully, I had somehow overlooked actually starting the IDS software. We were nearly blind to possible break-in attempts, because our IDS had not been running.

The critical flaws were in neglecting the pre-existing non-administrative user base entirely, and not bringing our intrusion detection system online promptly. Although we didn’t achieve a leading position at the end of the competition, the experience was invaluable. I learned chiefly how to authenticate users from disparate services against a central user base, and not to neglect administrating those users’ accounts. They do not take care of themselves. I also learned how to install and configure a commonly used intrusion detection system on a network gateway (and firewall) to help log and detect suspicious network activity. I also learned how to send logs from various machines to a central logging server.