CCDC 2019 Writeup
Pat Heaney, Evan Eastwood, Jared Butterfield, Namo Ziegler
Rahul Emani, Petr Esakov, Cole Daubenspeck, Brant Goings

Machines Given w/ Services (add versions where applicable):

1. Solaris w/ 22 and Wordpress 80
2. CoreOS w/ 22
3. GhostBSD w/ 22 and Nagios 80
4. Ubuntu w/ 22 and NFS 2249
5. Photon w/ 22. Nextcloud 80, and RocketChat 8080
6. Windows Server Core 2016 w/ LDAP
7. Windows Server 2012 w/ SMB 445, RDP 3389, and IIS 80
8. Windows Server 2008 w/ SMB and MSSQL 1433
9. Windows XP SP3 w/ SMB and MYSQL 3306

Individual Team Member Thoughts:

Solaris - Jared

- Pre-Comp Practice
    - Setup Solaris in a VM with Wordpress running, locked it down with firewall rules and ModSecurity
    - Researched how to uses Solaris PF firewall
    - Learned most useful Solaris commands
- Competition Day
    - Added a separate user and added them to /etc/sudoers, removing the admin account
    - Found a user that seemed suspicious, sshd, removed it, however it was needed for SSH
    - Removed two crontabs and disabled cron, one crontab was running a binary executable file
    - All Wordpress files were owned by the sshd user
    - Put firewall rules in place for ports 22, 80, and ICMP ping
        - Also disabled IPv6 by blocking all IPv6 connections in pf.conf
    - Spent 5 hours unsuccessfully trying to get SSH and HTTP to work
    - Finally got SSH to work, but the scorebot couldn't access it because it was on a different subnet and had no access to the default gateway
- Post-Comp 20-20 Vision
    - Should've thought of there being no default gateway
        - Makes sense now as we could connect in our LAN through SSH but the scorebot couldn't access any services
    - Should've learned more Solaris commands as I Googled a lot of them

CoreOS - Petr

- Pre-Comp Practice

- Competition Day
- Post-Comp 20-20 Vision

GhostBSD - Evan

- Pre-Comp Practice
    - Learned how to use BSD systems
    - Learned how to stand up nagios XI
    - Research CVE-2018-8733
- Competition Day
    - Change passwords
    - Setup IPFW rules
    - Attempted to get SSH functioning properly
    - Attempted to secure nagios from CVE-2018-8733
    - Attempted to stay alive for 5 hours while using the dead OS known as GhostBSD
    - Attempted to troubleshoot strange log in issue
- Post-Comp 20-20 Vision
    - I did not learn enough about the way Nagios Functions on the host machine
    - I did not learn enough about SSH settings in depth
    - I did not properly configure my written IPFW scripts

Ubuntu - Rahul

- Pre-Comp Practice
    - Downloaded various versions of Ubuntu and set up NFS and SSH on them
    - Attempted to attack the systems with my Kali VM
    - Learned to prevent blank logins and secure ssh configuration
    - Learned secure NFS configuration to secure mounting the file systems
    - Learned user management and looked through stigs
    - Set up a windows server machine to practice in event of windows support being necessary
    - Learned UFW rules in order to setup firewall
- Competition Day
    - Set up SSH config to secure against malicious exploiting of SSH
    - UFW rules were set up but were not working correctly
        - Disabled firewall completely
    - NFS was not working and tried to troubleshoot as best I could
    - User management was difficult
    - Crons were disabled or turned off by me
        - Thank god, since it was set up to take down my SSH and NFS daemons every so often
- Post-Comp 20-20 Vision
    - SSH was up for most of the time
    - NFS oddly went down and troubleshooting it was impossible

    - User management was not done as early as I had hoped, since red team was attacking within one minute of the competition starting
        - Frustrating
    - UFW was not working as specified which is very confusing
    - Trying to understand why UFW rules were so ineffective

Photon - Cole

- Pre-Comp Practice
    - Practiced with Photon V1 minimal
    - Enabled pre-installed SSH client: SSH 2.0, OpenSSH 7.1
    - Enabled default Docker version: 1.11.0
    - Downloaded RocketChat 0.57.1 from DockerHub store with ltest MongoDB image from DockerHub
    - Created docker-compose.yml file to setup containers
    - Created iptables rules (minus Docker chains) and tested them
    - Tested locked down SSH configuration
        - Allow incoming to only critical services
        - Allow only related, established outgoing
- Competition Day
    - Locked down SSH
    - Created iptables rules
    - Locked unauthorized users
    - Created script to alert on successful logins attempts
        - Didn't work in competition, but worked in testing
    - Restarted Docker to rebuild iptables chains
        - I have no idea how Docker got its configuration to restart automatically
    - Regularly checked SSH logs for login attempts
        - root login authenticated, but was blocked by SSH configuration
            - changed root password, then Red Team gave up
- Post-Comp 20-20 Vision
    - iptables rules should have been set up after SSH and accounts were locked down to reduce the window of successful attacks
    - Docker should have been restarted immediately after setting up iptables so it would rebuild the rule chains.
        - Not doing this caused unnecessary service downtime

Core 2016 - Brant

- Pre-Comp Practice
    - Research basic AD Powershell commands
    - Research possible GUI solutions to the headless system
- Competition Day
    - AD was filled with enabled users but we weren't given a list of “valid” users
    - Guest was enabled and had administrator rights

        - Disabled any administrators other than the administrator account
        - Firewall lists were extremely long
            - Disabled any IPv6 rules
            - Disabled “jenkins” rule along with any other odd looking rules
            - Disabled odd services through task manager
            - Monitored established connections through TCPView
    - Post-Comp 20-20 Vision
        - Should’ve researched powershell more so I didn’t rely on Google as much
        - Should’ve been more familiar with the Windows F/W and how it ran traffic against rules and in what order that happened

2008 and XP - Pat

- Pre-Comp Practice
    - To practice, I set up the XP and 2008 VMs in my home lab environment. Ran through patching, malware scanning, and firewall+user account lockdown in short time frame. Practiced managing MySQL server management via CLI. We requested a tool to help manage MySQL via a GUI, but the install didn't work; it was good that I practiced using MySQL CLI and reviewed sql knowledge (IST 210).
    - Windows Server 2008 set up MSSQL server and management studio, attempted to manage as well for practice.
- Competition Day
    - AD and local machine had many enabled users but we weren’t given a list of "valid" users.
    - Guest admin enabled, remote access group granted
    - xAdministrator existed and was continuously recreated, added to remote access
        - temporary solution while I worked on the real solution was to write windows task schedule items to run `net` commands to disable or delete accounts and stop malicious processes I saw
    - Firewall rules kept being reset and windows firewall completely disabled; was unable to find the cause during competition
    - c share kept opening
    - Monitored and killed processes using sysinternals tools
- Post-Comp 20-20 Vision
    - practice detecting existing malware and remediation more
    - would have behooved us to practice simulating red team activity more

2012 - Namo

- Pre-Comp Practice
    - Spun up a complete environment of 4 Windows servers in an AD domain at home for practice & testing
    - Studied IIS/OS secure configuration
- Competition Day

    - Crapton of malware pre-installed, pretty much unkillable without tools, runs even in safe mode, borks management tools and our firewall
    - As a result of that same malware, there was a lot of wonky stuff going on:
        - xAdministrator keeps popping back up with full admin rights
        - C:/ was always shared as “open” with full permissions to anyone - “net share open delete” didn’t help as the malware probably put that back soon afterwards
- Post-Comp 20-20 Vision
    - Should’ve remembered `bootrec /FixMbr, bootrec /ScanOs, bootrec /RebuildBcd`, the antidote to the Nyan Cat MEMZ trojan
    - Should’ve researched and practiced more on malware detection and removal

Injects:

1. Create a website: Successful
   a. Create a HTML file with a list of all the team members
      i. Red team actually deleted our “website” (likely through that “open” share), but we have the page loaded in memory and it was validated anyway
2. Create two users and test login: Successful
   a. Create users through Core 2016 and enable on 2008, then test login
3. Modify Nagios to monitor two open ports on Core 2016: Failed
   a. Nagios wasn’t working at the time
4. Create RocketChat user accounts and post intro in the chat room: Successful
   a. Console access in RocketChat UI to create accounts and post
5. Change Solaris hostname to io: Successful
   a.
   ```
   /etc/hosts
   /etc/nodename
   /etc/hostname.<interface>
   ```

Debrief:

Overall, the team did an excellent job of communicating within their own operating systems. The Linux guys helped each other out by testing F/W rules against each other along with SSH access. Windows guys helped each other through domain access and AD setup.

In the future, be prepared to be flooded with information from the get go. We barely looked at our requested downloads the entire time because we were consumed by the vulnerabilities in our individual systems. If you setup a list of priorities that you feel are important, stay vigilant and work through those without getting lost in other tasks. F/W rules, open ports, and default creds should be very high on the list.

Understand that Red Team will win. This year our Windows boxes were bricked in the last half hour by a looping Nyan cat gif in the MBR.

Red Team AMA:

- Solaris
  - route add default 192.168.3.254
- Round 3 was hard because Saturday and more time to share info
- Major screw ups
  - Web app code
  - Default creds
  - SSH sessions stayed open
- What defenses worked well?
  - Sysinternals
- Malware
  - MYSQL Procedure (RUNCMD) - runcmd.dll on XP
- White Team was looking for very detailed MR
  - Hashes
  - Exact location
  - Everybody's MR were bad
- Red Team used Mimikatz quite a bit