ANALYZING THE RED TEAM KILLCHAIN

By
Dennis
Devey

Things We Didn't Do Well

- Availability and **Compliance**
  - We lost most of our points here, as always
    - Ridiculous things being turned off in AD, I got to listen to white cell mocking us
    - Mostly just CDX being annoying
- We got BURNED on DNS A C2 out of the Linux workstations
  - Single process ran the whole time from initial callback
- Windows firewall misconfigured
  - Totally off on one workstation, and verbose error message on the other allowed them to ID our defensive measures

Things We Did Well

- Windows Firewall Rules
  - Broke their initial backdoors and first attempts at exploitation
  - Made Mudge write a stager in assembly for x64 internet explorer
- Didn’t have any loose credentials
  - They never got domain creds
- Network was secure
  - No lost services
- As always, CDX comes down to Grey Cell, not network defense
  - We could have not lost a single point from Confidentiality and Integrity and we still probably wouldn’t have won.

BLUF:

Key Takeaways:

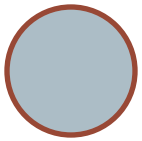
- All compromises were via Grey Cell
- All compromises were via Cobalt Strike
  - No matter the attempts at mitigation, beacons called home eventually
- Nearly all post-exploitation activity was done using Powershell

Mitigations:

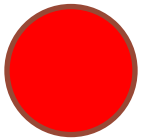
1. Sandbox and run all content sent to Grey Cell, pre-emptively block callback domains
   1. https://blog.rootshell.be/2012/06/20/cuckoomx-automating-email-attachments-scanning-with-cuckoo/
2. Use Cobalt-Strike as your threat actor
   1. Expect beacons to be successful
3. Disable Powershell

Slide Key

- Kill Chain

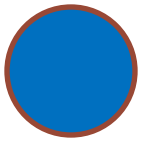


- Red Team



  - Tools
  - Techniques
  - Procedures

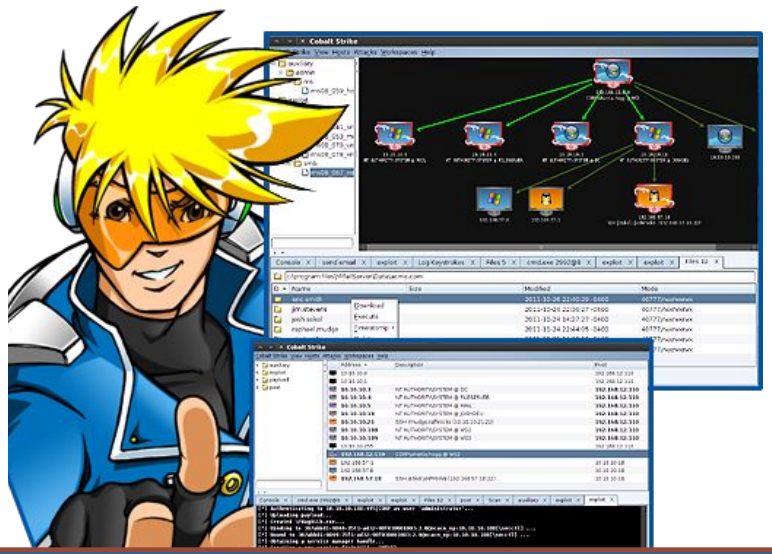
- Mitigations
  - Endpoint
  - Network



Cyber Killchain

- **Step 1: Reconnaissance.**
- **Step 2: Weaponization.**
- **Step 3: Delivery.**
- **Step 4: Exploitation.**
- **Step 5: Installation.**
- **Step 6: Command and control.**
- **Step 7: Action on objectives.**

Windows Red Team Tools



Cobalt Strike

Ripat

pupy

Poison Ivy and other RATs

Tools

- Cobalt Strike
  - Their number one tool, idiot proof Windows exploitation ©
  - 90 % or more of activity
  - Author is the key player on Red Cell's Windows Team
- Ripat
  - Red Team home-brew RAT, uses http beacons
  - Used for long term persistence, especially in injects and the compromised workstation images
    - **Did not** work against us because of Windows firewall rules
- Azala Linux Rootkit

Other Tools

- Powershell Empire
  - Link to Empire
  - I predict it will be the most powerful Windows tool by next year, if not the next month and a half
    - BUT
      - We can simply block it by disabling powershell
- Powersploit
  - Link to PowerSploit
- Poison Ivy
  - Famous RAT
- Pupy
  - Link to Pupy
  - Ultra modular python based open source RAT
  - Occasional use, people's pet projects
- Metasploit
  - Only would be used by attackers who don't know any better

1. Reconnaissance

How do they figure out what your network looks like?

TTP's

- The rules of CDX tell them what boxes you have, and what is run on them
- No scanning allowed until Monday at 18:00
- Most scanning occurs at night, between 22:00 and 06:00
- The next morning, there are beautifully formatted pdfs of every schools' networks
  - Fail to see most hosts on network

TTP's

- Nmap
  - Hosts and Ports
- Nessus
  - Vulnerability Scanning

Backdoors

- Pre-installed backdoors on workstations
  - Malware, mostly RIPAT
  - I didn't hear about any listening shells
  - Priv. Esc. 'stuff'
  - Fork bombs (I wouldn't expect to see this one again)

- The backdoors are off and non-functional until someone turns them on
  - Turned on by Red Cell, **not time based**, they debated when to do it
    - Hmmmm…..
    - Does that mean they are beaconing?

Mitigations

- Have everything patched
- Have correct ports closed

- La Brea Tarpit / some alternative
  - Return every port as open when scanned
- Temporarily blacklist scanning IP's
- **Honeypots!**

2. Weaponization

How is the malware created? What does it look like?

Networking TTP's

- One click DNS setup to register new domains
- One click IP redirect setup

Remember, they can stand up infrastructure in minutes(actually, they're not that good. Hours) , way easier for them to keep making domains than for you to detect them

TTP's

- Executables and other nasties built with Cobalt Strike Artifact Kit
  - Malleable artifact profile
  - Malleable C2
  - Host(s) to call back to is predefined when created
- C2 channels redirect to actual CS team-server

Mitigations

- Run every single possible Cobalt Strike attack, see which ones work

3. Delivery

How does malware get onto the system?

TTP's

- "Hey Grey Cell, can you download this and send it to yourself?"

100 % of All Compromises Via Grey Cell Email

- Vast majority .exe's
- A few .hta's

Mitigations

- Pull executables and links from email and throw them into a malware analysis VM, execute them, black hole domains they try to call back to
  - If this is automated, we will be able to pre-emptively block domains before the actual one is executed
  - https://blog.rootshell.be/2012/06/20/cuckoomx-automating-email-attachments-scanning-with-cuckoo/

4. Exploitation

How does the initial exploit occur?

TTP's

- "Hey Grey Cell, can you double click on that"
  - Log all of the everything

Mitigations

- Monitor process spawned from executable
  - Look for connections outbound
- Look for injection and injected processes
- Monitor file accesses (flag.txt)
- Monitor persistence mechanisms

5. Installation

How does the malware maintain persistence?

TTP's

- Countless ways to establish persistence
  - 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
  - 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)
  - Run persistence in whatever tool you're using
- Most common I saw was scheduled tasks
  - On startup
  - On program load
  - On a time
- Many many other ways to do it

Mitigations

BASELINING!

BASELINING!

BASELINING!

- File system changes!
- Registry changes!
- Settings changes!
- Processes/services changes!
- User account changes/creation!

But seriously, baselining.

6. Command and Control

How does the malware communicate?

Oh there are a lot of ways.

And remember folks, writing signatures for http and https is a losers game.

Red Team C2 MO

- Get first callback
- Inject into important process

- Establish long haul (30 min or greater) callback for persistence

- Create burnable beacons for short term actions
  - If I catch one beacon, I should catch the whole damn family

Cobalt Strike

- The Infamous Beacon
  - HTTP
  - HTTPS
  - DNS
    - A Records

Luckily for us, this is the most well documented piece of malware in the world, Thanks Raph!

http://blog.cobaltstrike.com/

Subscribe!

BLOG

Aggressor Script's Secret mIRC Scripting Past

April 6, 2016

Aggressor Script is the scripting engine in Cobalt Strike 3.0 and later. If you want to learn more about it, I recommend reading the documentation. In this blog post, I'll provide some history around Aggressor Script so you can better [illegible]tand it and where it comes from.

The mIRC Factor

mIRC is a popular client for Internet Relay Chat. In the mid-nineties, I [illegible]f a community of enthusiastic computer users who would interact with [illegible] online. Through this community, I had mentors and I was exposed to [illegible] on as well. mIRC was more than a GUI client to connect to IRC though. [illegible] also a programming environment. User scripts could create new IRC co[illegible] (aliases), respond to events, and even modify the presentation of mIRC's [illegible] This gave power users a lot of room to make mIRC their own.

How did we use this power? It depends on the user. Some would write the[illegible] scripts to express their artistic prowess. mIRC became their canvas, Cp4[illegible] characters their brushes. Others would write scripts to task multiple mIRC insta[illegible] (clones) to send messages to a friend that elicit an automatic response. Th[illegible] friend's automatic response to all mIRC instances would cause the IRC server to disconnect them for malicious flooding. These flooding games, among friends of course, were a popular use of mIRC scripting.

The jIRCii Factor

Welcome...

Welcome to the Cobalt Strike blog by Strategic Cyber LLC's Raphael Mudge

Contents

» Adversary Simulation
» Announcements
» Armitage
» Cobalt Strike
» Interviews
» metasploit framework
» Red Team
» Strategic Cyber LLC
» Uncategorized

[illegible]ubscribe

[illegible] Posts
[illegible] - Comments

[illegible] your email address to [illegible]out about new posts by [illegible]il. I won't use your email for any other reason.

Enter your email address

Subscribe

RIPAT
(Remote I____ P_____ Access Tool)

- HTTP beacon, usually deployed as their long term backdoor
  - What they put in the workstations
  - Injects as well

- Very slow, very stealthy. Not very functional, used to load Cobalt Strike modules

Windows Firewall DOES NOT Let It Out

Other Windows RATs

- Myriad of HTTPS, HTTP, and DNS A
- Poison Ivy, Empire, Throwback, Meterpreter, Mayhem, Kraken, De-security . . .
- Nothing too crazy, but we can't *totally* count out them developing one that visits a website and pulls from there
- Raph also plans on making an ICMP one
  - We have that blocked so it doesn't matter

Linux RATs

- Myriad of HTTPS and DNS A

- We got destroyed by DNS A out
  - Priority for blocking next year

General Mitigations

BASELINING!

BASELINING!

- Alert on all never before seen domains

But who needs that because…


ULTRA TOP
SECRET

Operation Beacon Eater

This is dumb but it will work. Worth a look at the overall idea of fingerprinting malware C2 servers because they all look pretty similar and behave the same for the same families.

Operation Beacon Eater

1. Red Cell creates new domains for beacons to call home to

1. Each piece of malware calls home to multiple domains

1. **NOT ONE** operator sets up an .html page for their newly created domains

Every single Rubberneck request goes to an actual page which loads in your browser

If you visit a C2 channel address, it either 404's or times out…

What does that mean?

Any IP/Domain that 404's or times out can be treated as a C2 domain

- No matter the protocol, all we have to do is strip out the address visited, whether that is an IP or a Domain, and then visit it ourselves with a script
- If it returns a web page, it is benign traffic.
- If it 404's or times out, it is a C2 channel, and we add that ip/domain to a list
  - Iterate through this list with a parallel and identical script to the one that populates it.

But how do we know its not just the network failing?

Short answer, we don't but here's how we deal with that:

By maintaining a list of domains that fail our check, and iterating through them, we can have a constantly updated blacklist.

Anything that fails we're not getting points for anyway, soooo nothing of value was lost

Other Mitigations

- FLOW ANALYSIS

- Look at number of unique requests per domain
  - For http, https, dns

- All the things we did this year
- So many other things we could do
  - Writing rules for indicators is a losers game, but it *can* work

7. Action on Objectives

- What actions does the adversary take once they have access?

Red Team Lateral Movement MO

- Establish persistence
- Elevate privileges
  - ‘bypassuac’, ‘elevate’, powershell Powerup
- Dump hashes, run mimikatz
- Hope they have admin creds
- Otherwise they just sit there and hope someone logs in with domain creds
  - “Snake pit” traps, they break something and make a domain admin come fix it

Red Team Token Stealing MO

- **No Privilege Escalation Required!**

- Run native windows commands
  - Confidentiality
    - “type token.txt”
  - Integrity
    - “echo ‘redteamwuzhere’ > token.txt”

Some scripted, most manual. Only needs to be done once every 6 hours.

Elevate Privs

- Try to make them spawn in low integrity processes
  - Kinda difficult.
- Perhaps figure out how to identify the escalation attempts and kill process.
  - Sounds like a capstone
- Don’t Allow Powershell

Dump Hashes and Mimikatz

- Monitor lsass

- Ton of research to be done

Domain Cred Control

- Every time you log in with domain credentials on a box, ASSUME COMPROMISE

ASSUME COMPROMISE

Monitor Token Access

- Kernel level magic
  - Sounds like a capstone

Powershell

- Don’t Allow Powershell
  - We certainly talked about not allowing it…. But we did anyway.
  - Next year, absolutely not.
    - And if we can get the process ID that tried to use Powershell? That’s even better than blocking it outright.

We CANNOT Allow Them to Use Powershell Next Year

CMD Line

- Disable CMD line for all non-admin users
  - They were really annoyed with that on other teams

Red Team Nastiness

- Evil tricks, didn't do too many this year
- Mostly because everyone's networks already were terrible

Metasploiters

MS08_067

MS09_050

Other than that…. You're probably good.
I didn't see anyone even load it up while I was there.

DoS

- I saw a few DoS, not much we can do about that except identify and call them on it the next morning

Website Defaces

- Every single website was owned the same way
- Happens to be the same way they do it every year

- Using curl they send commands to “somewhere” on our servers, where they are executed as root
  - Research required

Snake Pits

- Detailed write up eventually.

Overwriting MBR

A favorite red team trick.

- There is no good way to defend
- There is really no good way to recover
  - Only seen one team who recovered with a CD
- You must revert.

Questions?

OTHER LINKS

I will integrate all of these into the actual presentation, eventually.

- Anything Labled “Red Team Debrief” is excellent
  - Just google for them
- http://lockboxx.blogspot.com/2015/03/red-teaming-at-prccdc-2015.html
  - Very worthwhile step through of their ttp’s
- Red Team Field Manual
  - The Bible
- Red Team Wiki
  - Plenty of ttps
- My ill gotten gains
  - Even more ttps

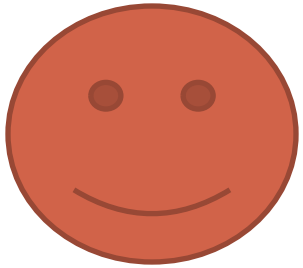
Raph Mudge- Mr. Cobaltstrike

- Dirty Red Team Tricks
  - https://youtu.be/oclbbqvawQg
- Dirty Red Team Tricks 2
  - http://blog.cobaltstrike.com/2012/10/04/dirty-red-team-tricks-ii-at-derbycon-2-0/
- Auto Hack Scripts
  - http://www.fastandeasyhacking.com/dirty
- http://blog.cobaltstrike.com/2014/03/04/ccdc-red-teams-ten-tips-to-maximize-success/
- http://blog.cobaltstrike.com/2016/02/23/cobalt-strike-tips-for-2016-ccdc-red-teams/
- http://blog.cobaltstrike.com/2013/04/24/national-ccdc-red-team-fair-and-balanced/
- http://blog.cobaltstrike.com/2015/04/17/so-you-won-a-regional-and-youre-headed-to-national-ccdc/

More Mudge

Honestly, his whole site is great stuff. I will distill it down, but its all good.

http://blog.cobaltstrike.com/



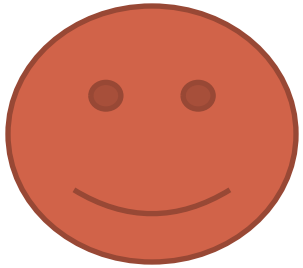
HOW TO WIN CCDC

A Red Team perspective

First created in 2010, Updated each year. Last update 2/3/2016

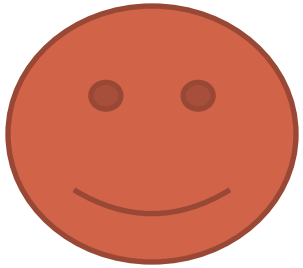
https://docs.google.com/presentation/d/1pPXLg3KqwSMLRCNRfows5QnVI2mLjSmll5vN2WHMFJg/mobilepresent?slide=id.ga0cc710_3_41

Firewall admin



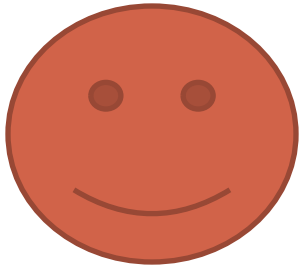
- RAISE SHIELDS Mr Sulu!
- Monitor OUTBOUND connections
- Know your firewall and how to configure it
- Have or know exactly where to get any and all software you need to administer the firewall given to you.
- Egress and Ingress filtering
- IPv6 OFF (Unless required)
- deny any any is your friend
- Wireless gear is your baby, WPA2, WPS off (if possible), and long pass phrase
- Pass off Incident Reports to IR person
- CAPRICA (ACL generator) is _AWESOME_
  - http://code.google.com/p/capirca/

Linux Admin



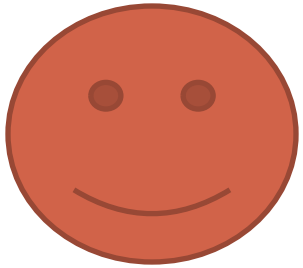
- Upgrade your kernel ASAP
- Fail2Ban
- If ($PHP) then shoot.self; (Fix php.ini)
- SETUID
- Watch those auth logs
- Create a process list file so IR can diff it
- Remove any unused users or services
- IPTSTATE is like TCPview for Linux, use it. love it.
- GRSEC IF YOU HAVE TIME, custom kernels take time to compile but, it's fun to watch Red Teamers attempt privilege escalation on older kernels.
  - Turn off the ability to change grsec settings via sysctl
  - Turn on EXEC logging
  - Watch the audit log for signs of escalation attempts

Linux Admin (cont'd)



- File Integrity logging pays dividends:
  - Tripwire
  - OSSec (has pre-configurations for most *nix)
- Nothing new should enter here without you knowing:
  - /tmp/ (new files or binaries in here are bad news)
    - .hidden directory is a common place to put stuff
  - crontab for all users
  - ~/.ssh/ (and /root/ not just /home)
  - /etc/
  - /etc/passwd & /etc/shadow & /etc/sudoers
- Know all SetUID binaries and watch for new ones

Linux Commands



- Final all 'immutable' files
  - find . | xargs -I file lsattr -a file 2>/dev/null | grep '^....i'
  - 'chattr -i file' to change it back
  - Doing this on / takes a long time, point it where it counts: /etc/, ~/, /tmp/ etc.. etc..

Sorry Raph.. :-)

time find / | xargs -I file lsattr -a file 2>/dev/null | grep '^....i'

----i---------- /etc/bob.txt

----i---------- /etc/bob.txt

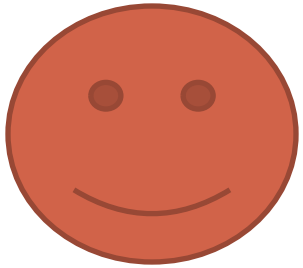
real 9m15.451s

user 0m51.505s

sys 6m38.862s

Just /etc => real 0m2.674s

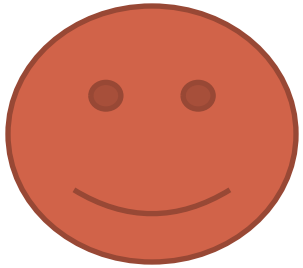
Windows Admin



- Event Viewer is your friend
- Autoruns is your friend
- Process Explorer and TCP View are your friend
- OSSEC works for windows too
  - (agent only, must talk to a Linux server for reporting)
- Change passwords and fast! (Automate if possible)
- Remove unused users and services
- Turn your firewall on and REMOVE EXCEPTIONS
- Turn off Teredo

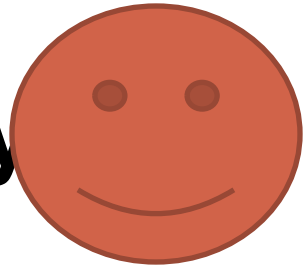
Mark Russinovich is your friend.

Windows Admin - Changing Passwords Fast



- Program one:
  - AutoIt (make a binary to do it faster)
- Download one:
  - http://bit.ly/bulkpasswordcontrol (AD only - not local)
  - Advantage: pseudo random passwords
- Built in one:
  - dsquery user ou=Users,dc=testlab,dc=net | dsmod user -pwd RedTeamSucks! -mustchpwd yes
  - LAPS for local admin passwords (Not built in, but it is Microsoft tool) https://technet.microsoft.com/en-us/library/security/3062591.aspx

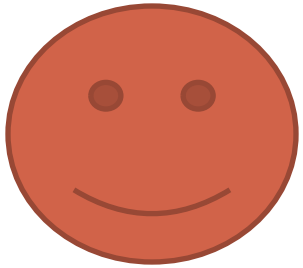
Windows Admin - GPO (Security



Some specific Windows Group Policy to set Security Options

- Network security: LAN Manager authentication level - Send NTLMv2 response only\refuse NTLM & LM
- Network security: Do not store LAN Manager hash value on next password change - Enabled
- Network access: Do not allow anonymous enumeration of SAM accounts and shares - Enabled
- Network access: Do not allow anonymous enumeration of SAM accounts - Enabled
- Network access: Allow anonymous SID/name translation - Disabled
- Accounts: Rename administrator account - Rename to something unique (but remember it)
- Interactive logon: Message text for users attempting to log on - sometimes an inject

Windows Admin - GPO (Audit)

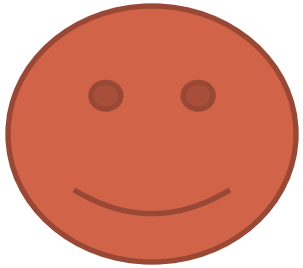


Audit Policy

Learn to configure windows audit logs and understand the events.

- Audit process tracking - Successes
- Audit account management - Successes, Failures
- Audit logon events - Successes, Failures
- Audit account logon events - Successes, Failures

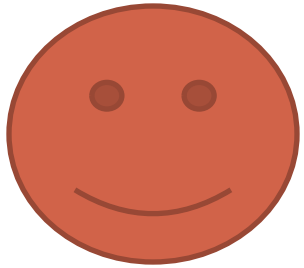
Windows Admin - GPO (Other)



User Rights Assignment

- Debug programs - Remove all groups/users
- Allow log on through Terminal Services - Leave blank to disallow login via TS even if it has been started.

Windows Admin - Local GPO



Local GPO is much faster to push out on small networks, and can be applied to any Windows system, not just domain joined ones (plus if the attacker kicks a box off the domain, domain GPO goes away). There isn't an easy way to do it for all GPO settings, but for security ones 'secedit' is your friend.

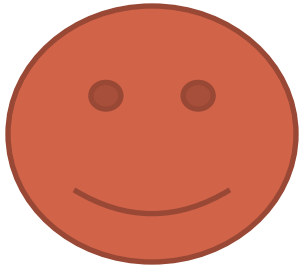
-- Export a config from a VM or other default install for reference:

secedit /export /cfg checkme.inf

-- Edit to to have more secure settings then import onto your target system:

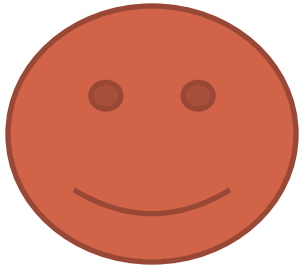
secedit /configure /db secedit.sdb /cfg securecheckme.inf

Web Admin



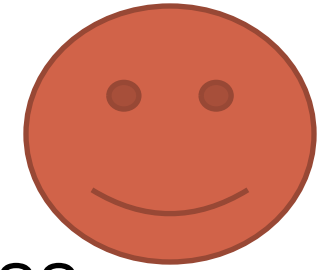
- Mod_Security
  - (get the linux admin to install it quickly, and get comfortable installing it on Windows)
  - http://blog.spiderlabs.com/2013/04/web-application-defenders-cookbook-ccdc-blue-team-cheatsheet.html (just ignore the honey traps portion, you normally won't have time to set or monitor for them)
- Passwords… find them, reset them, most likely the Red Team found them first
- Look for administrative interfaces and restrict them to localhost or an "admin" box

Know the Red Team tools



- Run Poison Ivy, know how to remove it
- Run Metasploit's attacks psexec, MS08_067, and MS09_050 and see what changes are made to the system
- Run Metasploit's persistence script, know how to get rid of it
- AUTORUNS is your friend

Other Resources



- http://ambuships.com/ <- Free HIPS that kicks ASS
- https://github.com/trustedsec/artillery <-- Sorta another HIPS but both Win and Linux
- http://la-samhna.de/samhain/ SAMHAIN - Linux IDS / File Integrity monitor
- OSSEC...
- Lynis (Linux security checking)
- https://www.trustwave.com/Resources/SpiderLabs-Blog/Web-Application-Defender-s-Cookbook--CCDC-Blue-Team-Cheatsheet/

More Resources

- http://www.blackhillsinfosec.com/?p=5368
- https://github.com/adhdproject/adhdproject.github.io/blob/master/index.md
- http://www.slideshare.net/scriptjunkie/red-teaming-the-ccdc
- https://www.wraysec.com/2016/03/14/how-to-win-the-ccdc/

Tools

- https://github.com/byt3bl33d3r/CrackMapExec
- https://github.com/enaqx/awesome-pentest

Programs

- Cobalt Strike
  - Cortana
- PowerSploit
- Empire/EmPyre

- PowerTools
- Mimikatz
- Bloodhound
- PowerSCCM

- https://blog.netspi.com/15-ways-to-bypass-the-powershell-execution-policy/

Things to Use

- Microsoft's Sysinternals
- File Auditing
- Microsoft Baseline Analyzer
- https://github.com/lchack/CCDC
- https://github.com/mike-bailey/CCDC-Scripts
- CredentialGuard