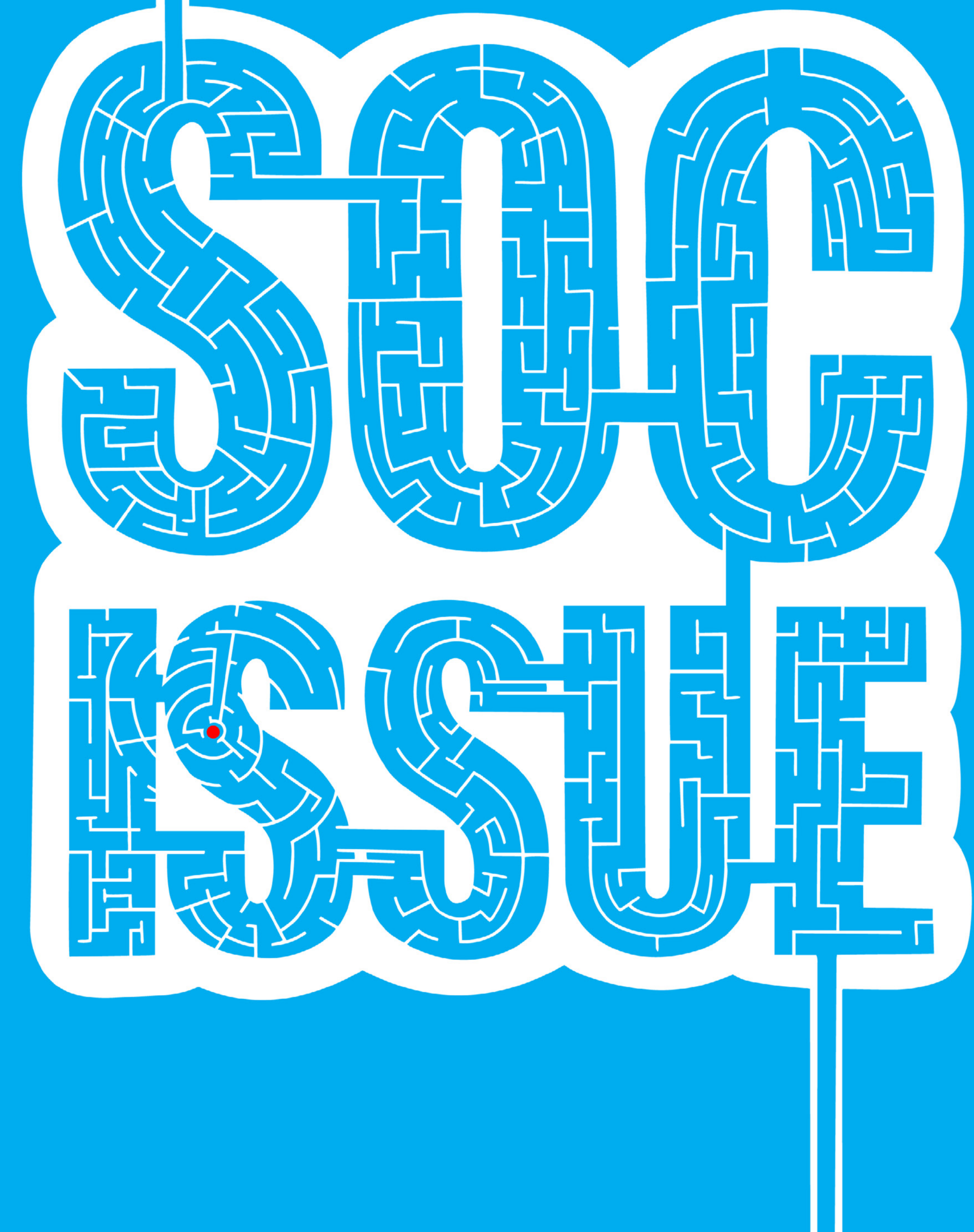
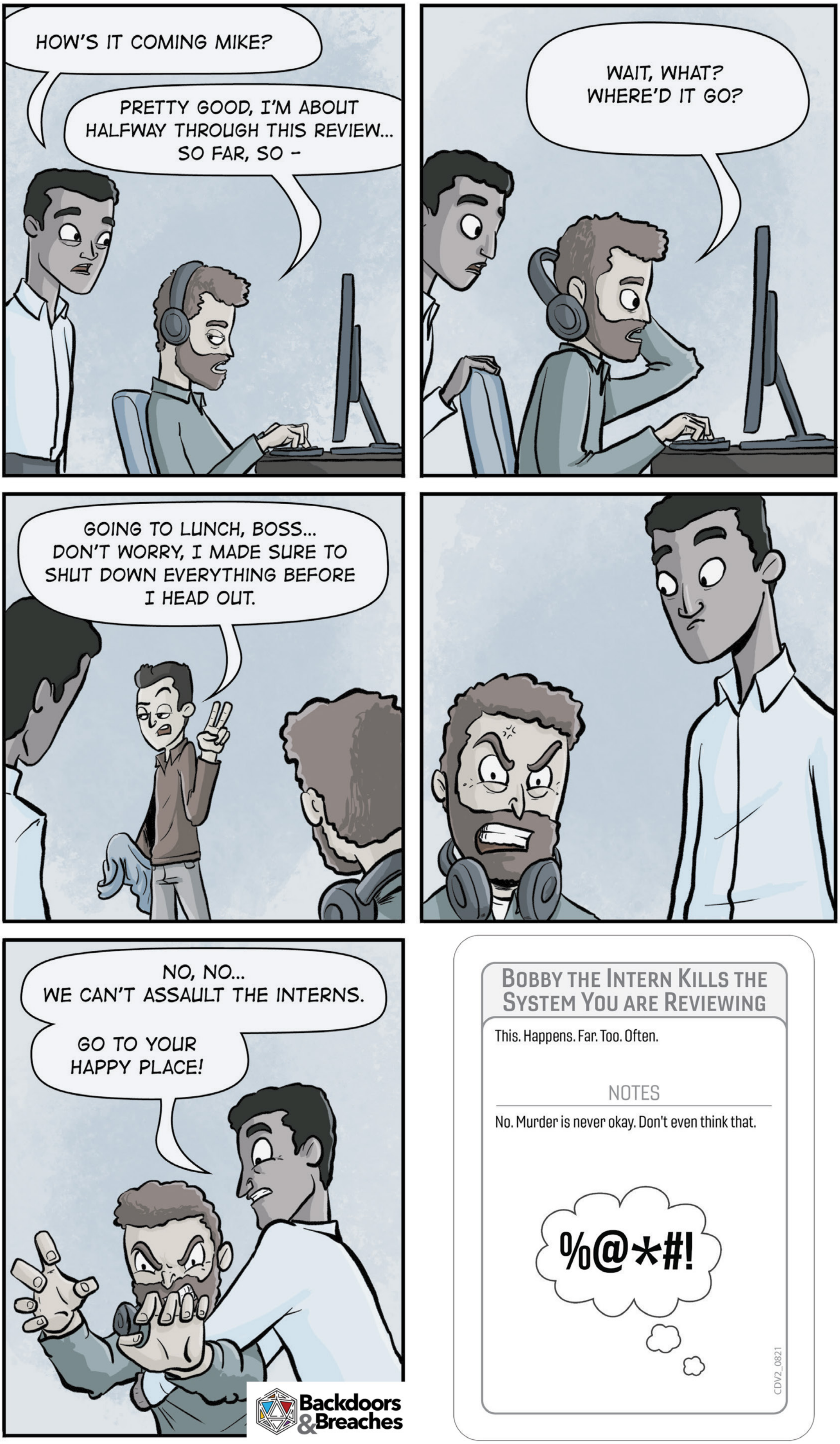
PROMPT#



BY BRADLEY SAMUELSON


HOW'S IT COMING MIKE?
PRETTY GOOD, I'M ABOUT HALFWAY THROUGH THIS REVIEW... SO FAR, SO -
WAIT, WHAT? WHERE'D IT GO?
GOING TO LUNCH, BOSS... DON'T WORRY, I MADE SURE TO SHUT DOWN EVERYTHING BEFORE I HEAD OUT.
NO, NO... WE CAN'T ASSAULT THE INTERNS.
GO TO YOUR HAPPY PLACE!
Backdoors & Breaches
%@*#!
CDV2_0821

Bobby the Intern Kills the System You are Reviewing

This. Happens. Far. Too. Often.

NOTES

No. Murder is never okay. Don't even think that.

PROMPT#

SOC ISSUE

CREDITS

PUBLISHER
John Strand
@strandjs

EDITOR-IN-CHIEF
Jason Blanchard
@BanjoCrashland

DESIGNER & ILLUSTRATOR
Caitlin Cash
CaitlinCash.com

EDITOR
Kassie Kimball
linkedin.com/in/kassiekimball

ASSOCIATE EDITOR
Andrea Culling
andreamculling99@gmail.com

PROOFREADER/MORALE OFFICER
Deb Wigley
@debthedeb

PUZZLE MAKER
Alex Minster
@Belouve

BACK COVER ARTIST
Eddie "the y3t1" Mize
deviantart.com/eddiethey3t1/gallery

SNOOT MODEL
East the Dog
@troopthedoggo

We like to make things.

Moreso, we like to make things for YOU, our community. Without your support and feedback, the previous issues of PROMPT# wouldn't have been possible. Your kind words have fueled us through all the difficult parts of this process.

This time 'round, we decided to narrow our theme. We asked various members of our Security Operations Center (SOC) team: "What do you think are the most important aspects every SOC should focus on?" Then, we roped them into writing about it. But don't worry, even if SOC isn't your thing, there's still lots of fun and knowledge to be found within these pages!

I sincerely hope you've enjoyed what we've made for you.

~ Kassie Kimball, aka @sneaky_ _space

ARTICLE CONTRIBUTORS

Noah Heckman
Dave Hoff
William Stearns
Keith Chew
Max Boehner
Kaitlyn Wimberely

SPECIAL THANKS

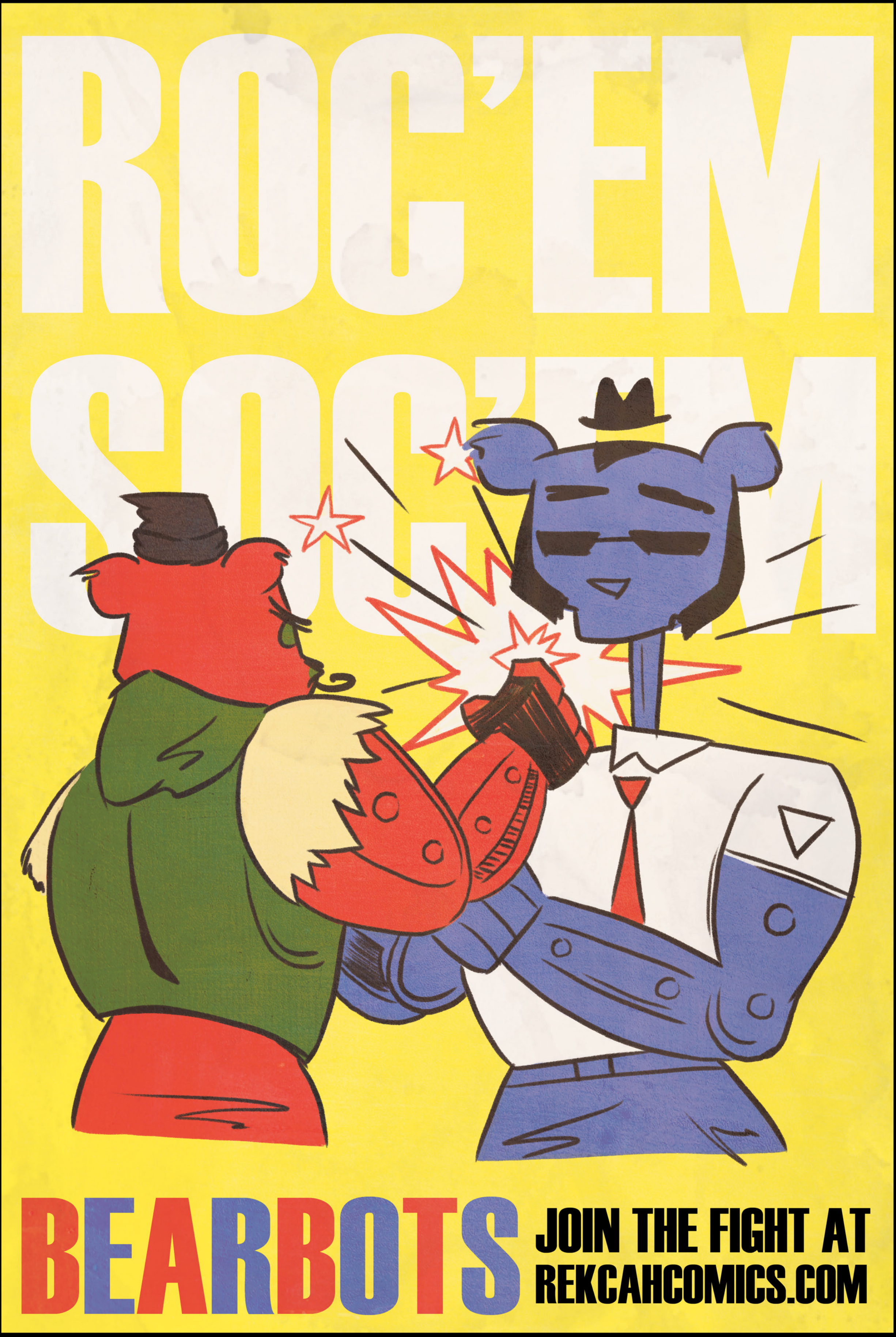
To the team at BHIS and the members of our awesome community, like YOU!

OTHER CONTRIBUTORS

Amanda Berlin
Olaf Hartong
Anna Hughes
Bradley Samuelson
Matt Thomas
The Traveler
Princess Peaches the Dragon

BLACK HILLS
Information Security
115 West Hudson St.
Spearfish, SD 57783

Contact Us: 701-484-BHIS / info@promptzine.com
PRINTED IN CANADA. POLYBAGGED IN CANADA. CARDS PRINTED IN USA.


ROC'EM
SOC'EM
BEARBOTS
JOIN THE FIGHT AT
REKCAHCOMICS.COM

INTO THE DEEP BLUE

BY NOAH HECKMAN

Breaking into cybersecurity is not easy. If you're reading this, you probably know what I'm talking about. From the first time I was asked to assist the cybersecurity team (while I was just a helpdesk employee with a lot of ambition), I knew I was going to push myself into this industry or grey my hair trying. The reality is that I did a little of both. As bleak as the road may seem at times, I encourage you to keep trying. These are some of the things I learned on my journey into security.

To start, "being in security" is not just a job; it's a mindset. If your goals are to work for companies with passionate security personnel, that mindset is critical. It doesn't matter if you are working at a helpdesk, as a developer, or even doing something unrelated to IT at all. Being in security means you don't say that "someone else will report that phishing email" or "it's not my job to mention that this service is way outdated." You don't stand by and watch as your coworker who lost their keys breaks into the office with a credit card slipped behind the latching mechanism. Calling out stuff like this does two things. First, it establishes correspondence with the security team, which creates name recognition and over time, may mean they remember you when they're looking for a new analyst. Second, it fixes legitimate issues that would endanger your company and its assets… which is the whole point!

Looking back to when getting into security was my only goal, I can't help but think about how much time I spent working. And not only for my employer, but also for myself in the lab. Anything I wasn't allowed to do at work, I would just do at home. (Don't have domain admin privileges at work? No problem, just spin up a domain controller and attach some workstations to it… but it's a long-term lab, so I should have redundant DCs, right? Now just multiply that attitude by a thousand.) I had a full dev environment of my company running in my basement on a couple of old Dell servers. Which is fantastic. Do it. You will learn a ton from stuff like that. However, there is a catch.

I was spending 8+ work hours a day patching, automating, and watching for security issues in the SIEM. And then I would go home and do the same in my lab for another few hours. The way I saw it, my company wasn't giving me the opportunities I wanted, so I made them for myself... and yes, there are some issues with this. For one, do put your lab experience on your resume, but don't be surprised when companies don't even glance at it. If you are going to do it, then you do it for your own betterment, not for your employers'. On that same note, don't give away time to your workplace. If you are supposed to be working a 40-hour week, stick to it. Those late nights finishing up a project because you "don't have anything better to do" will contribute to burnout all the same. Stop eating your dinner at the keyboard. Or your lunch, for that matter. How can you do your best if you don't set aside time for yourself?

You may ask, "Well, if I do that, how will my work get done?" To which I would have to ask why you are the only one on your team who knows how to do that work. If you are **not** the only one, then congratulations, there's your answer. However, if you are the only person in your team who knows how to do specific things, I would strongly encourage you to reach out about cross-training. Pretty much any IT course, training, or consultant will tell you how important it is to have redundant systems. Why is it any less relevant that there be redundant personnel? To some, it may be a matter of job security. If that's the case for you, you're only accomplishing one thing: ensuring that no junior members can fill your role and you will always be in that position, limiting everyone from moving up within your company. Long story short: if you have a team, use them. Share your knowledge with each other and I promise you will learn and accomplish more than you ever could on your own.

I came to BHIS from the Discord, serving as a Nerd Herder (which is hands down one of the best roles I have had the pleasure to hold). If you don't know, the Discord server is filled with tons of people sharing information with each other while expecting nothing more than you are willing to give in return. The Nerd Herders are a group of community members who donate their time to help make our community better. That includes providing support during training classes, helping new members understand what we are all about, and even showing up on "the news" from time to time. Several do not hold official security jobs but, in my mind, every single one of them are cybersecurity professionals, shaping the industry into a better place. If you are not part of the Discord, I recommend you check out what it has to offer.

So, what happens when you finally make it? You are in a new security role at a new company. It's smooth sailing from here, right? All that imposter syndrome you have been holding inside you is now validated and gone, yeah? In my experience, that's not the way it goes. It wasn't until I realized that I was holding a spot that more qualified individuals wanted, that I realized what true imposter syndrome was. But despite the inner voices telling me that I was just "taking up space" and "preventing better employees from getting stuff done," I concluded that I needed to use that to drive myself. The fact is, I genuinely love my work and what I do. And the other fact is that I do prevent someone else from having this job. If for any reason I decide to perform my duties with anything less than my very best, then I owe it to those who would to take over my position. The moral is: Don't let imposter syndrome get you down. Let it drive you to be the best that you can be.

If you take nothing else from my account, hear this: It may be hard now, but remember this time. At some point when you make it, turn your attention behind you and offer someone below you a hand. Remember the gatekeepers, the bad managers, and the close-minded interviewers that you ran into along the way and make a point to not be like them. Because for some of you, there will be a day when you are a hiring manager, and you can make the choice to either give someone a chance or to keep perpetuating the same issues you are currently facing now. Remember this time.

TOP 3 SKILLS YOU NEED TO START IN SOC

BY OLAF HARTONG

AN INQUISITIVE MIND

The need to understand is a vital skill for working any job in security. Without desiring to comprehend the why, the how, the when (and so on), you'll struggle to make proper judgement calls. Investigating these factors doesn't always require a full technical approach. Thankfully, there are a lot of awesome people sharing their useful and accessible research online.

There are many forms of creativity. There are endless ways of finding a different angle or reframing context to tell a compelling story. By being creative in your approach, you'll often end up with results you might not have expected upfront but end up providing great value.

THE ABILITY TO TAKE A STEP BACK (OR GET HELP)

While investigating an incident, it's common to go down a rabbit hole and get stuck on something, somewhere. The ability to pull back and rationalize what you are looking at, what you are looking for, and how that all fits into the bigger picture will give you new avenues of approach or provide you with the contextual means to know when to ask for help.

If you cannot find the information yourself, never be afraid to seek support or let someone share their knowledge with you. Nobody knows everything, and being part of and encouraging a culture of open sharing will benefit everyone.

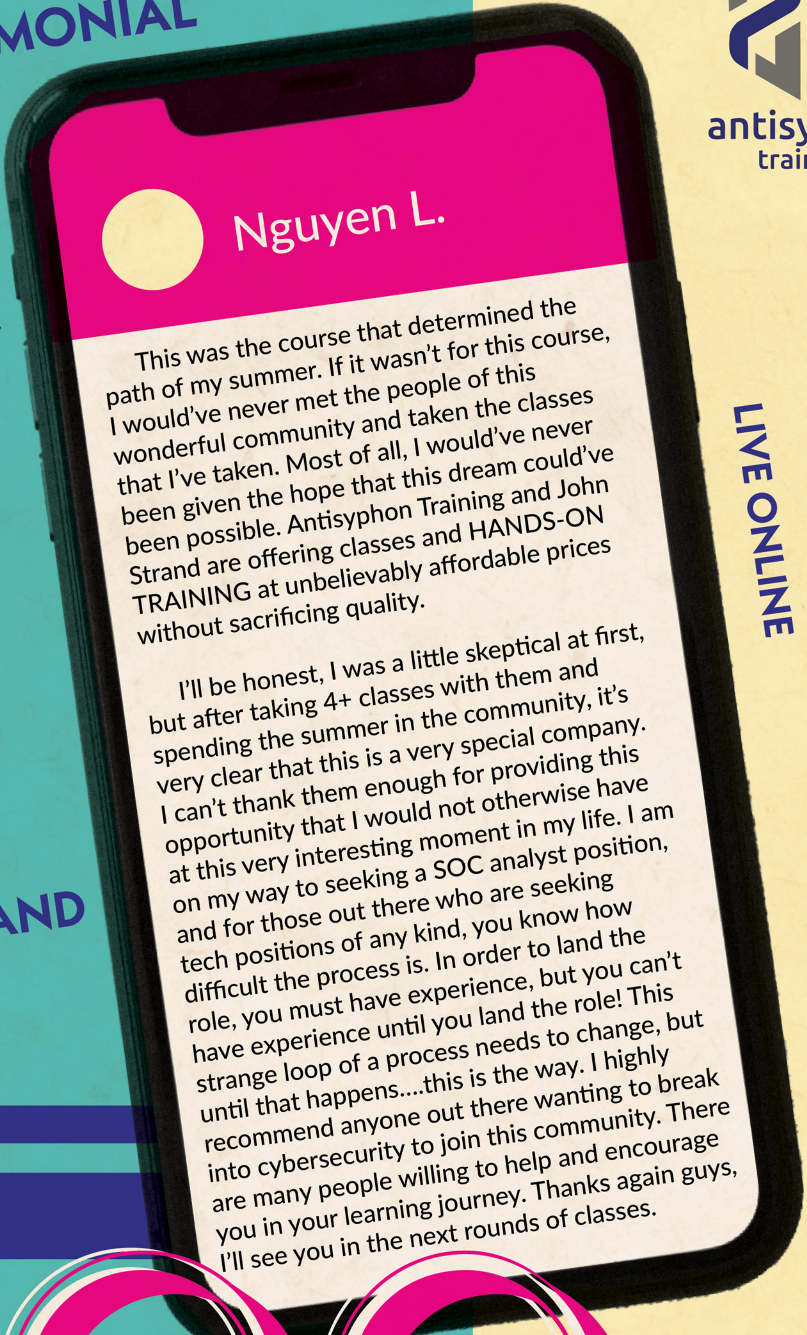
TECHNICAL INFRASTRUCTURE UNDERSTANDING

Having (and continuously building) a deeper understanding of how operating systems work and how enterprise networks are built will benefit you greatly in your day-to-day. This understanding of what is normal for the OS, as well as why legacy systems and certain implementations exist (usually due to all kinds of business or cost reasons), will make sure you can properly analyze and respond to events.

Additionally, you will have the confidence in either making the proper assessment or in your ability to explain your thought process to someone else when asking for support.

ACTUAL STUDENT TESTIMONIAL

antisyphon training



Nguyen L.

This was the course that determined the path of my summer. If it wasn't for this course, I would've never met the people of this wonderful community and taken the classes that I've taken. Most of all, I would've never been given the hope that this dream could've been possible. Antisyphon Training and John Strand are offering classes and HANDS-ON TRAINING at unbelievably affordable prices without sacrificing quality.

I'll be honest, I was a little skeptical at first, but after taking 4+ classes with them and spending the summer in the community, it's very clear that this is a very special company. I can't thank them enough for providing this opportunity that I would not otherwise have at this very interesting moment in my life. I am on my way to seeking a SOC analyst position, and for those out there who are seeking tech positions of any kind, you know how difficult the process is. In order to land the role, you must have experience, but you can't have experience until you land the role! This strange loop of a process needs to change, but until that happens....this is the way. I highly recommend anyone out there wanting to break into cybersecurity to join this community. There are many people willing to help and encourage you in your learning journey. Thanks again guys, I'll see you in the next rounds of classes.

HANDS-ON LABS

ON-DEMAND

LIVE ONLINE

antisyphontraining.com

SOC CORE SKILLS

PAY-WHAT-YOU-CAN TRAINING

LOG INGESTION

BY DAVE HOFF

Designing and setting up a new SIEM can be intimidating. While software vendors often provide support for cluster setup and sizing, setting up how you ingest logs is usually up to you. It's easy to feel lost in the wilderness without a map, especially if you're trying to bring in data from less common sources. Let's walk through a high-level plan of attack.

The first and most important aspect of log ingestion is choosing what you want to log. The most common mistake I see from companies designing their own SIEM is trying to log everything from everywhere. While there are benefits to having every log from every service you use in a single location, the cost to do so is high in both setup time and hardware/cloud expenses. Is keeping firewall logs in your SIEM valuable enough to justify reducing overall retention time to stay within budget? Your chances of success are much higher when you start by determining which log sources are most likely to contain useful, actionable information and focusing your efforts on those sources. Additionally, many individual log sources can be configured to reduce the amount of data they send. Whether this is done by upping the severity threshold (choosing not to ship "debug" or "info" level logs) or by excluding specific event IDs (common with Windows event logs), trimming down the amount of data sent by each service can have a large impact on your SIEM's performance and usability.

Once you've determined which logs are essential, it's important to plan out how to organize your data. Pick a naming scheme for your tables/indices that allows for future expansion while maintaining your sanity. For smaller organizations, you might be able to name an index with only the source service name, while larger organizations might split up their logs by department first, then by company, and lastly by service (i.e., infra-Amazon-CloudTrail). This is a step that often doesn't get much thought at the beginning but can be difficult to change down the road.

So, you've chosen your sources, have a great naming scheme, and your first log sources are sending data. What's next? Data normalization. Depending on how well your log source integrates with your SIEM software, your data will either be split up into nice, individual fields, or it will all be lumped into one big "message" field that will need to be parsed out. Either way, it's important to work towards common field names for as much of your data as possible. Take this situation as an example: Your Azure AD logs have a field named "azure.user.name" in the format "Domain\Username" but your Windows logs use the field "winlog.user_name" having only the username, no domain. Correlating data between the two sources will require your analysts to remember the differences and manually adjust their queries when switching data sources. If you're using Elasticsearch, you can map your data to Elastic Common Schema fields, which would indicate both above fields should be placed in the field "user.name" and the domain placed in "user.domain". Now you're able to punch in one query to search all your datasets.

Lastly, think about how you will be using the data you've ingested. Look for any common, repetitive actions your analysts take when searching the data. Do your Windows logs have the full path of an executable that's been run, but your analysts often use a leading wildcard to search for an executable name (*lsass.exe)? Do they frequently search for logs with an RFC1918 source IP to find outgoing connections? These situations can be improved with a concept known as schema-on-write. Instead of using these complicated queries (schema-on-read), it's often more efficient to split the executable path during ingest, creating an "executable.name" field that can be searched without wildcards. We could also automatically assign directions (outbound vs. inbound) to our network logs, making inefficient IP range queries unnecessary. Changes like these not only make your analysts' lives easier, they also can have a dramatic impact on search performance, especially when they reduce wildcard usage.

Log ingestion is a technical, complicated process and it's easy to get lost in the weeds while you're writing parsers and configuring log shippers. Sticking closely to a detailed plan will greatly improve usability, organization, and will ultimately save time and money over the course of the project.

INCIDENT RESPONSE

LEGOS

Incident Response (IR)

is not a flowchart.
Or, at least, it shouldn't be.

It's impossible to create a detailed flowchart of how you're going to handle absolutely every incident that can happen in your organization. There are far too many factors and variables in play.

Don't think in flowcharts or decision trees.

Think of incident response (and your core skills as a SOC analyst) as Lego bricks.

Instead of trying to have an IR flowchart, build skills you can utilize. Assemble your IR knowledge in a variety of ways to meet the challenge you're encountering. This allows you to react to a wide variety of types of incidents that can occur in your organization.

Your IR capabilities are flexible, sharp, pointy, and horrible to step on.

AN INCIDENT RESPONSE CARD GAME

Backdoors™ & Breaches

THE SOC

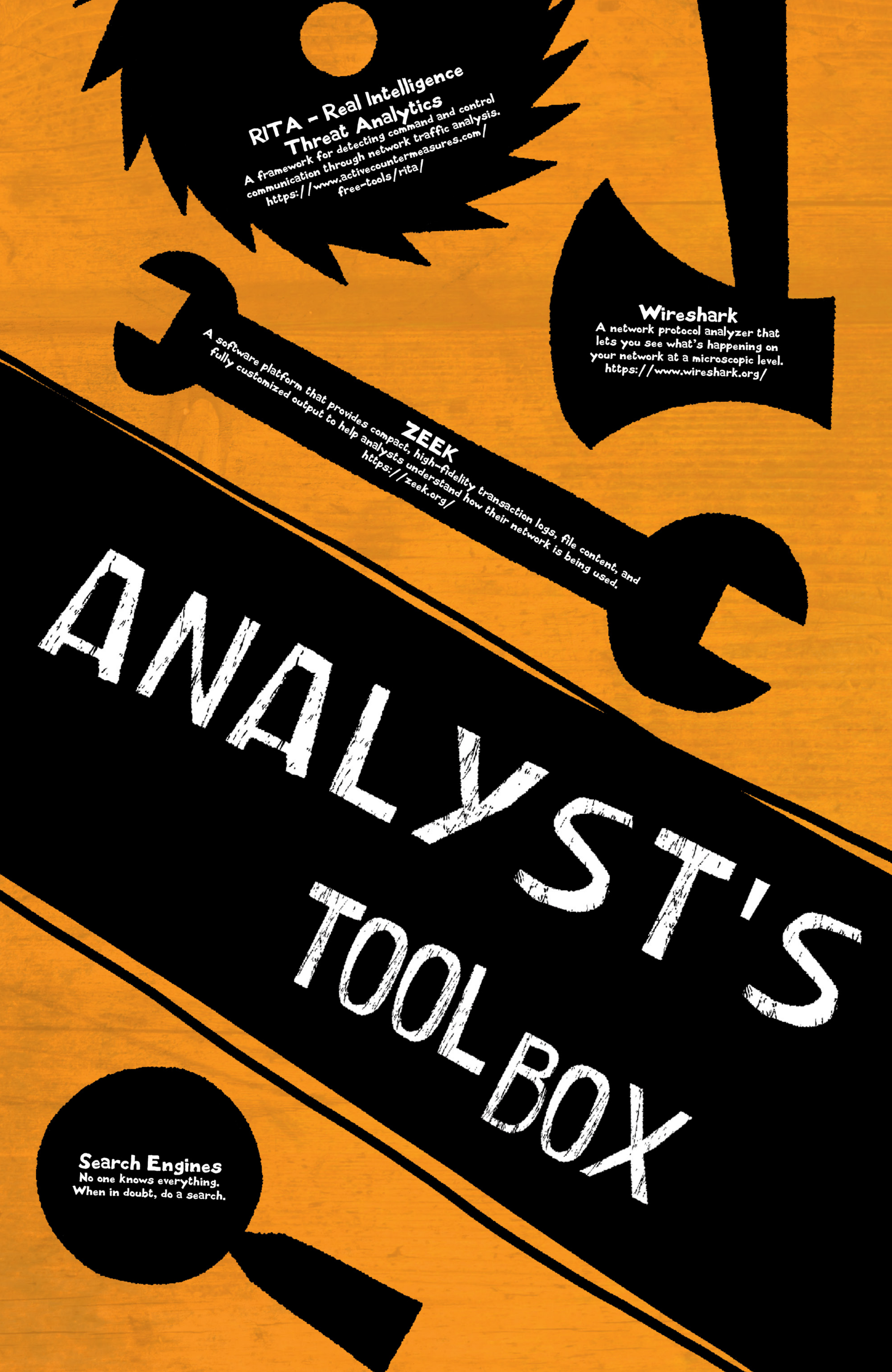
THT - Threat Hunting Toolkit
A Swiss Army knife for threat hunting, log processing, and security-focused data science.
https://github.com/ethack/tht

BloodHound
Uses graph theory to reveal hidden relationships within an Active Directory or Azure environment to identify and eliminate those same attack paths.
https://github.com/BloodHoundAD/BloodHound

Atomic Red Team
A library of tests mapped to the MITRE ATT&CK® framework that can quickly, portably, and reproducibly test your environment.
https://github.com/redcanaryco/atomic-red-team

DeepBlueCLI
A PowerShell Module for threat hunting via Windows event logs.
https://github.com/sans-blue-team/DeepBlueCLI

ANALYST'S TOOLBOX



RITA - Real Intelligence Threat Analytics
A framework for detecting command and control communication through network traffic analysis.
https://www.activecountermeasures.com/free-tools/rita/

Wireshark
A network protocol analyzer that lets you see what's happening on your network at a microscopic level.
https://www.wireshark.org/

ZEEK
A software platform that provides compact, high-fidelity transaction logs, file content, and fully customized output to help analysts understand how their network is being used.
https://zeek.org/

Search Engines
No one knows everything.
When in doubt, do a search.

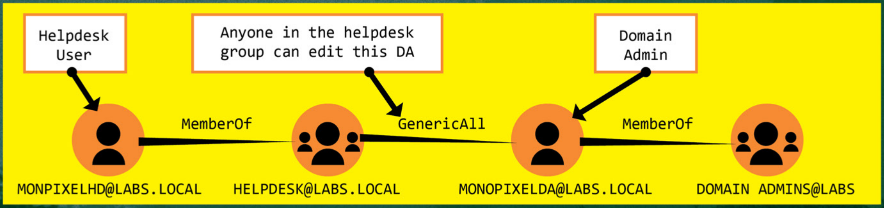
COMMON ACTIVE DIRECTORY PITFALLS THAT COULD RUIN YOUR DAY

By Noah Heckman

Active Directory is a critical part in many of our Windows networks. As the basis of trust, it makes an enchanting target for an adversary. However, on the blue team, we sometimes lose sight of it, since it "just works" in the background. Compiled below are a few common unsecure configurations that you should make sure don't exist in your environment.

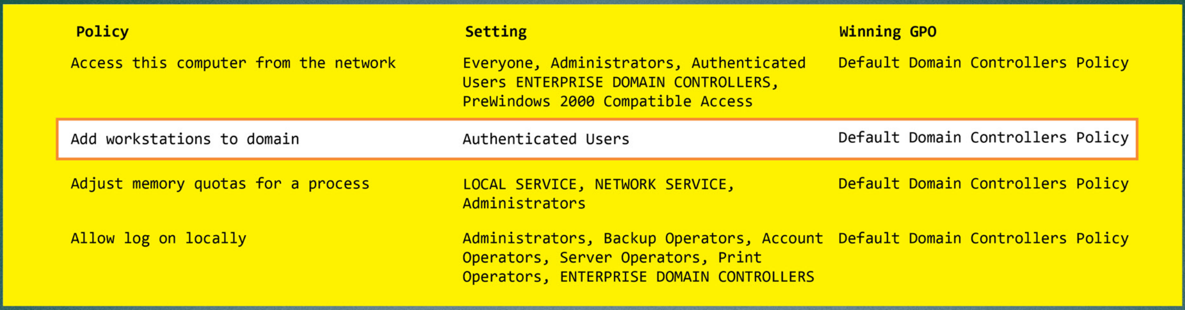
Control Paths: Who are your admins, really?

An extremely common issue I have found in Active Directory instances is that the Administrator groups indirectly contain more high-privilege accounts than the organization thinks are there. As an example, say an organization has proper account segmentation for their IT staff, such as one account for day-to-day emails and general business, one support account for basic admin tasks such as resetting passwords and making new low-privilege accounts, and one domain administrator (DA) account that is used for making new high-privilege accounts and other domain level tasks. Now, say that you allow all helpdesk accounts to reset all passwords. At first, this makes sense, until you factor in that this could also include your DA accounts. If a user can reset the password of a DA account, they effectively are a DA account themselves. These control paths are something that can be seen using tools like BloodHound (https://github.com/BloodHoundAD). Despite what many will say, Bloodhound is an incredibly useful tool to assist blue teamers in finding vulnerable control paths.



Pre-Windows 2000 Compatible Access: I doubt your computers are really that old, but...

The Pre-Windows 2000 Compatible Access Group allows its members to have access to legacy RPC calls that could be abused by an adversary. In Active Directory 2003, the Everyone group and Anonymous group are members of the Pre-Windows 2000 Compatible Access group by default. Having the Anonymous, Authenticated Users, and Everyone user groups as members this group effectively allows any user access to read all AD users and groups across the domain. So, if any user account was compromised, or if someone were to run malicious software, then the entire domain could easily be enumerated. Exploitation and enumeration of this legacy group has been seen in connection with the critical Windows print spooler vulnerability, PrintNightmare. Likewise, placing computer accounts in this group allows the use of legacy RPC calls on those member systems that, at a minimum, increases the attack surface of a system. As a systems admin, any devices, users, or groups in the Pre-Windows 2000 Compatible Access group should be regarded as less secure, and mitigations should be made to protect them.

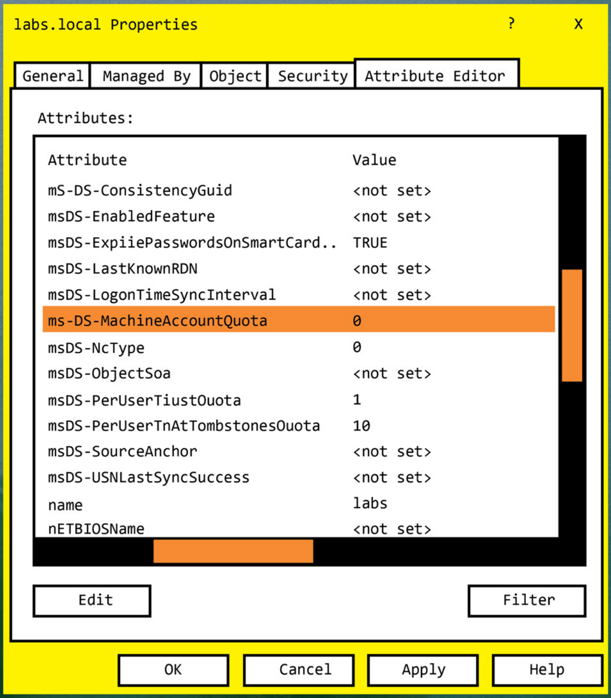


Policy	Setting	Winning GPO
Access this computer from the network	Everyone, Administrators, Authenticated Users ENTERPRISE DOMAIN CONTROLLERS, PreWindows 2000 Compatible Access	Default Domain Controllers Policy
Add workstations to domain	Authenticated Users	Default Domain Controllers Policy
Adjust memory quotas for a process	LOCAL SERVICE, NETWORK SERVICE, Administrators	Default Domain Controllers Policy
Allow log on locally	Administrators, Backup Operators, Account Operators, Server Operators, Print Operators, ENTERPRISE DOMAIN CONTROLLERS	Default Domain Controllers Policy

Free Admins! Authenticated Users Can Add Devices to the Domain

By default, the Default Domain Controller Policy GPO object allows the Authenticated Users group to add up to 10 computers to the domain.

The security concern of this is, of course, that it would be possible for an attacker to compromise a low-privilege account and then add a device to the domain that the attacker has full control of. This would eliminate the need to perform privilege escalation on a compromised system, or avoid the need to compromise a system at all if they acquire credentials out of band. Additionally, such devices may help an attacker remain undetected by providing bases of operations that appear like regular domain systems without containing any of the usual endpoint protection controls that might raise alerts. This also could potentially provide an avenue for attacks like we saw with CVE-2021-42278. Luckily, the fix for this is easy. First, we will want to restrict the above GPO to only allow IT staff to add devices to the domain. Next, we want to set ms-ds-machineaccountquota to 0 instead of the default value of 10.



Attributes:

```
Attribute                          Value
mS-DS-ConsistencyGuid              <not set>
msDS-EnabledFeature                <not set>
msDS-ExpiiePasswordsOnSmartCard..  TRUE
msDS-LastKnownRDN                  <not set>
msDS-LogonTimeSyncInterval         <not set>
ms-DS-MachineAccountQuota          0
msDS-NcType                        0
msDS-ObjectSoa                     <not set>
msDS-PerUserTiustOuota             1
msDS-PerUserTnAtTombstonesOuota    10
msDS-SourceAnchor                  <not set>
msDS-USNLastSyncSuccess            <not set>
name                               labs
nETBIOSName                        <not set>
```

To do this, just access Active Directory Users and Computer, and enable Advanced Features. Then, pull up the Properties pane for the domain and set the ms-ds-machineaccountquota to 0, as shown above.

In summary, BloodHound is your friend — even on the blue team — and you should run it on a regular basis to validate your control paths and group memberships. Tools such as Ping Castle are also free to run in your environment and find items such as the ability for users to add workstations. Ultimately, it is infinitely easier to correct these issues now, rather than after an incident.

10 FUN THINGS TO DO WITH RASPBERRY PI

BY WILLIAM STEARNS

The Raspberry Pi is a single-board, credit card-sized computer that runs Linux, has the normal ports you'd expect on a laptop, and costs around $50. They can be standalone systems for a particular project, servers, or desktop systems.

So, what can you do with one?

Create an NTP time server for your building with a GPS receiver

https://austinsnerdythings.com/2021/04/19/microsecond-accurate-ntp-with-a-raspberry-pi-and-pps-gps/

Capture packets from a mirror port on a switch

https://www.youtube.com/watch?v=_Ih_wjHafrM

Set up a flight monitor using Software Defined Radio

https://flightaware.com/adsb/piaware/

Create a large display to show network traffic and other statistics

https://www.google.com/search?q=raspberry+pi+network+monitoring

Share a Pi with a young relative or child and let them pick a project they'd like to do

https://learn.adafruit.com/search?categoriesLvl0=Raspberry%2520Pi

Block ads from getting through to your other systems

https://learn.adafruit.com/pi-hole-ad-blocker-with-pi-zero-w

Connect a Pi to a custom-built electronic circuit

https://learn.adafruit.com/search?q=GPIO&categoriesLvl0=Raspberry%2520Pi

Monitor your house

https://learn.adafruit.com/monitor-your-home-with-the-raspberry-pi-b-plus

Set up a portable security gateway/pen testing system/remote network monitoring system

-- Get a Raspberry Pi 4 with at least 4GB of memory and install your favorite Linux distribution on it (look for the "ARM 64", "aarch64", or "Raspberry Pi 4" flavor)

https://www.kali.org/docs/arm/raspberry-pi-4/

Use the Pi to have an inexpensive second system on your desk! Take a look at the Pi 400, especially a kit that comes with all you need except for the monitor for about $100.

https://www.adafruit.com/product/4796

If you didn't see a project above that interests you, take a look at the Raspberry Pi Foundation's website:

https://www.raspberrypi.org/

[Note: at the time of this writing, the Raspberry Pi's are hard to find because of the worldwide chip shortage. We hope that by the time you read this, that shortage has eased.]

HACKERS
IN THE MIRROR

We left a short message for you to read;
Discover the key, you have what you need.

Look for hackers in the mirror, if you hit a stop.
Decoded lines have a PROMPT#, linked to the top.

Use every bit, they all have their place -
Complete the grid, to show you're an ace!

Fill out the message, and you'll be gold.
Bugged by more? Well, there's more to behold.

One final layer, waiting for you:
Remember the top? That first tricky clue?

A final word, encoded in PROMPT#s, is what you'll find.
To solve it all, just think outside your mind.

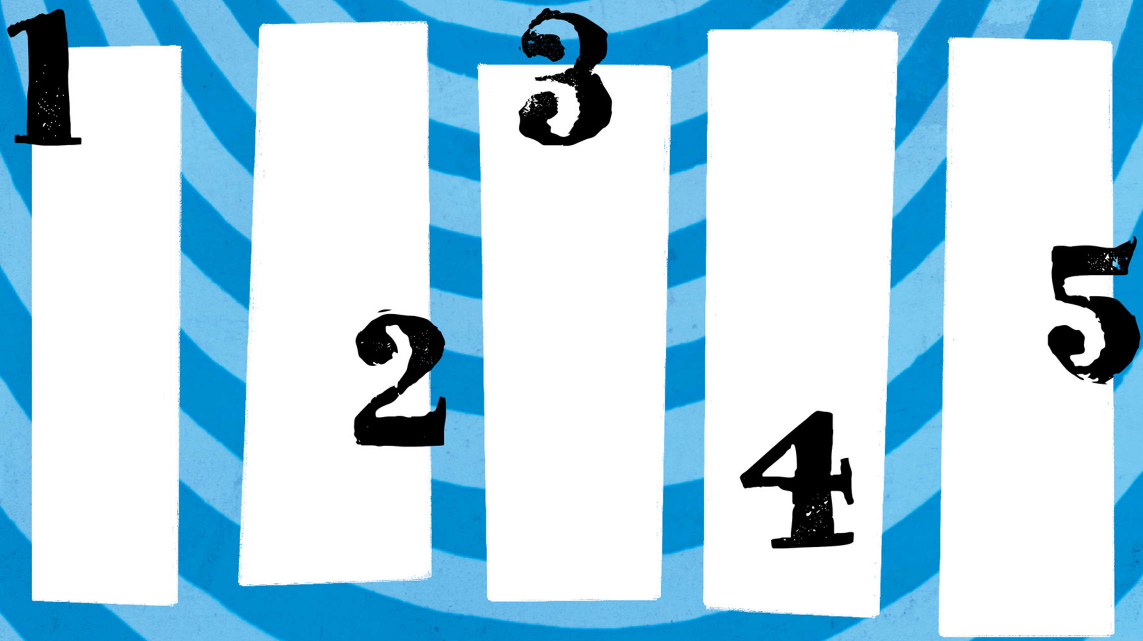
Save these keys, you may see them again.
Share them with others, your family, or friend.

Use it yourself, we'd love you to share with us!
As always, be creative and stay curious!

- Alex Minster

	*	8	¢	;	)	6	9	#	0	5	(
1											
2		E									
3											
4								B			
5											
6											
7				R							
8											
9											
10								T			
11											

JOHN STRAND's



5 PHASE PLAN

FOR STARTING IN COMPUTER SECURITY

PHASE 1

Learn your core operating systems. Build a lab. Get started with a language. Learn basic security fundamentals.

Start your education with the soft skills. Understand the technology: how are these machines used in business? What are people doing with them? You can be as technical as anyone, but if you don't understand the application of what you're trying to do and if you can't SPEAK THE BUSINESS SPEAK, you won't get far.

Windows:

Go to the Windows Evaluation Center. Install software from Microsoft. This is going to be painful. Some things are easy to install, like Active Directory. Some things are very, very difficult to install, like SCCM or Configuration Management. But these are important lessons for you to learn. Set up the things that you will be constantly defending (or constantly attacking) as a security professional.

Linux:

Install everything… from scratch. Don't know how? Visit a search engine. Type your question. Click the button. Don't give up just because it's hard. Security isn't about taking the easy route — it's about constantly learning, even under exceptionally difficult circumstances. The only way to get good is by struggling. If you need to, remove your easy way out and uninstall Windows.
Also, learn Bash scripting (there are other shells, but Bash is the one you're gonna end up using more than not).

Networking:

Set up a network lab. First, get your stuff at home up-and-running and make sure you KNOW what it is doing. Then, get some simulators (https://www.brianlinkletter.com/open-source-network-simulators/). Get some gear. You can buy old equipment for cheap on eBay. Take it apart. Find out how it works. Buy two or three of things… you're gonna end up breaking a few.

Coding:

Learn to code. Python is the best place to start (though other languages are important to learn). Study online. Code Academy, Code Warrior, and Pluralsight are all great resources, among many others.

Security Standards:

Learn the CIS top 20 Critical Security Controls. The AuditScripts Critical Security Controls Master Mapping spreadsheet (https://www.auditscripts.com/free-resources/critical-security-controls/) is an incredibly valuable resource. It can help you learn not only one framework, but directly apply that to a variety of other frameworks through its intensive cross-referencing. Knowing these is a big plus in your resume. It's strategic and high-level. Learn it.

PHASE 2

Time to start projects! (You may have already… that's fine!)

Move from being a consumer, to a creator

You should:

Start a security group (working on a team is an important experience)

- At work
- At school

Learn PowerShell (…this will take a while)

Keep up-to-date on security news

Eliminate distractions that are holding you back

PHASE 3

This is the time of web apps — you'll have to know these

Start with PHP and ASP.NET (don't get distracted by anything else yet)

Feel free to branch out to networked iOS and Android Apps

Learn to code (badly)

Develop SOMETHING

Dare to suck at something.

Embrace the suck.

It's okay.

PHASE 4

Time to start hacking stuff!

Learn IDA and Immunity Debugger

Pick a protocol and understand that protocol

Hit online challenges

(You've already been playing with Metasploit this whole time, right?)

Download ZAP from OWASP

Use and learn ALL this:

Windows ATT&CK for Enterprise Matrix

SANS Ultimate Pentest Poster

PHASE 5

PRESENT!

Give talks everywhere and anywhere

Present on things you JUST learned!

Take advantage of cons/events/webcasts as a speaker and…

Put. Yourself. Out. There.

IN CLOSING. . .

Feel free to:

- Indulge in distractions
- Stick to this plan
- Ignore this plan
- Develop your own plan
- Get good at just one thing
- Get a degree
- Don't get a degree
- Get certifications
- Don't get certified

Do NOT do the following:

- Sink into video games
- Waste your time figuring out the cube
- Binge watch shows on Netflix
- Use Bing for anything
- Just barely learn Metasploit to impress people
- Spend more time on the hacker "look" than learning
- Get angry
- Blame others

Check out the full video of John talking about his plan here:
https://www.blackhillsinfosec.com/webcast-john-strands-5-year-plan-into-infosec-part-2/

WORK CAN BE RUFF

IN CASE OF EMERGENCY...

BOOP THE

SNOOT
BLACK HILLS
Information Security

THE HUMAN ASPECT OF THREAT HUNTING

BY KEITH CHEW

We all have access to a plethora of network and endpoint security tools.

Many of these tools are designed to automate the security and defenses of our networks and devices by identifying signatures, processes, and methods of compromise that have been witnessed and discovered over time. We have our history documented, and these tools are effective at applying that knowledge towards protecting against similar events; but what about our future as attackers continue to innovate?

The question should be asked: “How did we discover these things that security software is designed to detect?”

I believe there are two answers to this question: 1) purely by accident, which in most cases is much too late... or 2) by actively threat hunting the network, leading to the discovery of the unknown. Barring the ability to mind read or otherwise predict the actions of those who would violate our spaces for profit, going forward, our mission should be to equally defend, actively seek, and eliminate as many network threats as we can.

This requires taking charge and implementing active human threat hunting. The most effective and holistic threat hunting starts with having a complete picture of the network and all the data. This entails collecting all network communication data arriving to and departing from the external perimeter of the network. Collecting all communications at the perimeter eliminates any potential blind spots created by internal network configurations, software, devices, switches, and/or routers. This provides you with records of all communications entering and leaving your network, without exception from all devices that can communicate between your network and the internet.

At a basic level, any compromised device (regardless of how it was initially compromised) needs to communicate to the outside world, either to receive instructions from the attacker or to exfiltrate data. It is through these communications and patterns we are attempting to identify threats that may have evaded security software detection by manual or automated means.

So, what are you looking for? As an analyst, there are many indicators of compromise to be mindful of and search for, but if it had to be boiled down to one thing, you are looking for anything out of the ordinary. An efficient threat hunter is one who is familiar with normal network activities that transpire on a daily basis, knowing what is ordinary and how to recognize anomalies. Examples of this could be, "Should the workstation used by Frank in HR be querying 2000+ unique subdomains of a root domain each day?" or "Should the postage meter used by the accounting department be establishing repeated short connections to an unknown host located in a hostile foreign country every 30-50 seconds?" Probably not. If you cannot justify a legitimate business need for these connections, why is it happening?

But... wow, a complete rolling network traffic capture generates mountains of data. How are you supposed to sift through all of this data to find potential threats? Surely none of us have the time to analyze network logs line-by-line. Fortunately, we all also have access to software tools such as RITA and AC-Hunter. Tools that analyze, correlate, and categorize raw network communications to continuously identify potentially malicious and anomalous network communication behaviors, can bring these communication pairs to the top for a human analyst to investigate and verify, or escalate as a further threat.

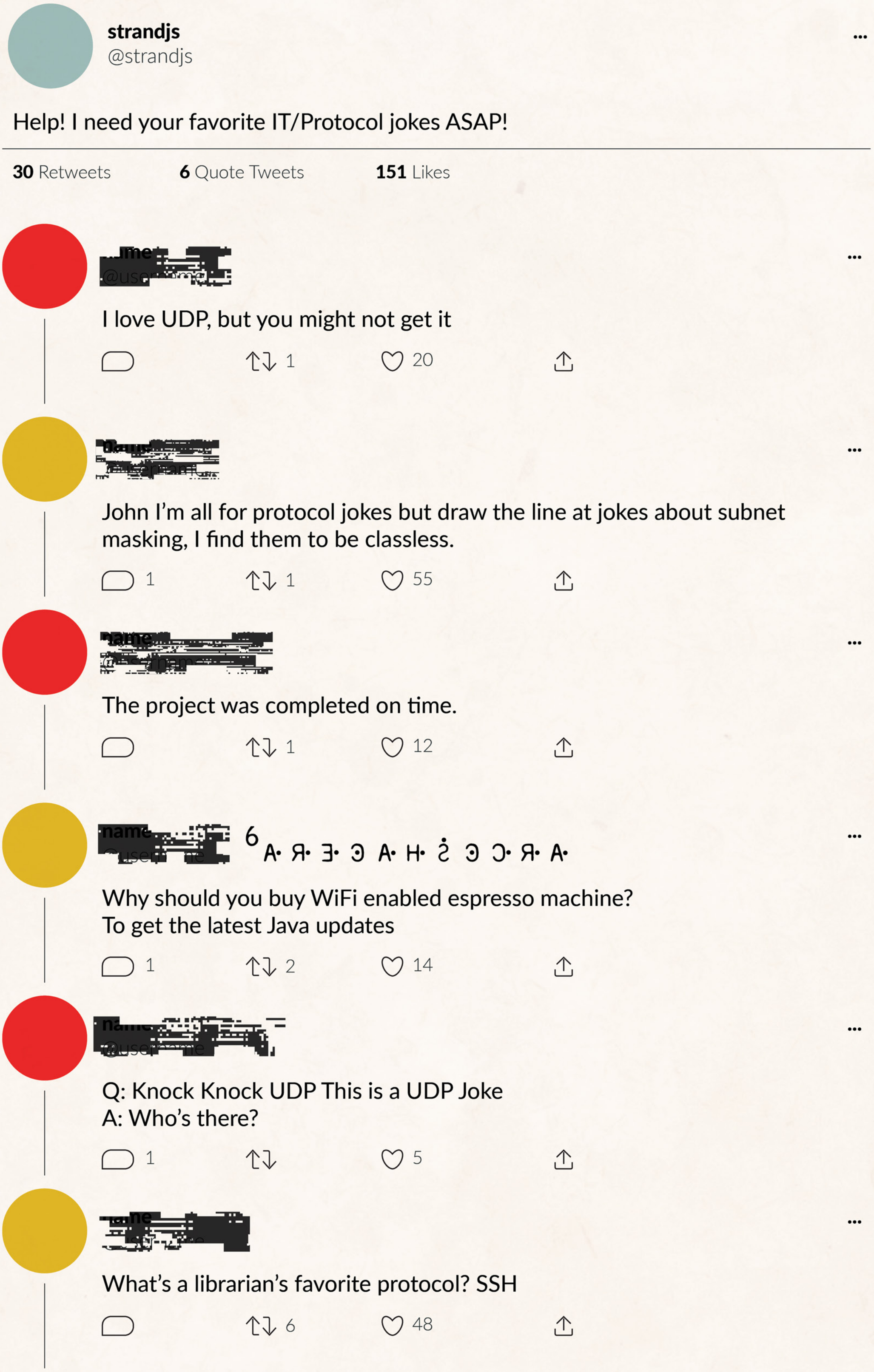
There are many valuable network security tools available to us. Wouldn't it be wonderful to implement these tools on our networks and then just sit back, waiting for a red light to start spinning if there's an issue, calling us to action? As much as we all want that, we are not there yet. We may never be.

A strong security posture requires engagement and is accomplished through a combination of effective security tools, automation, and active threat hunting by trained human eyes. This is the methodology we follow at Black Hills Information Security in our Active SOC and Hunt Team Operations Center.

Technology and tools are valuable assets, but the human analyst is invaluable.

HAPPY HUNTING!

JOHN'S GOT JOKES



strandjs
@strandjs

Help! I need your favorite IT/Protocol jokes ASAP!

30 Retweets **6** Quote Tweets **151** Likes

[redacted]

I love UDP, but you might not get it

↻ 1 ♡ 20

[redacted]

John I'm all for protocol jokes but draw the line at jokes about subnet masking, I find them to be classless.

1 ↻ 1 ♡ 55

[redacted]

The project was completed on time.

↻ 1 ♡ 12

[redacted] 6 A• Я• Ǝ• Ͽ A• H• ⸮ Ͽ Ↄ• Я• A•

Why should you buy WiFi enabled espresso machine?
To get the latest Java updates

1 ↻ 2 ♡ 14

[redacted]

Q: Knock Knock UDP This is a UDP Joke
A: Who's there?

1 ↻ ♡ 5

[redacted]

What's a librarian's favorite protocol? SSH

↻ 6 ♡ 48
]

MALWARE OF THE DAY

Gain confidence identifying threats on your network by checking out Active Countermeasures' Malware of the Day.

Be wary of malware, but don't be afraid!

Learn from the experts as they replicate and analyze real-world malware and attacker methods found "in the wild" to help you become more familiar with the network communication methods commonly seen from observed malware.

Check out the full library at:

activecountermeasures.com/category/malware-of-the-day

ACTIVE | COUNTERMEASURES

MENTAL HEALTH HACKERS

Q&A WITH AMANDA BERLIN

Q: WHO ARE YOU?

A: Hey! My name is Amanda Berlin, and I do a bunch of things :D I am the Lead Incident Detection Engineer at Blumira, podcast co-host on Brakeing Down Security, author of the Defensive Security Handbook, trainer at Antisyphon, speaker at random places and conferences, and, the reason we're all here on this page, the CEO of Mental Health Hackers!

Other than that cybersecurity stuff, I live in Ohio with 2 of my 3 boys. The oldest has flown away to work on F35's in the Marines, and the younger two are here with me, mostly playing Fortnite, but also going on outdoor adventures and playing board games.

Q: WHAT DOES MENTAL HEALTH HACKERS DO?

A: Our motto is "Hackers Helping Hackers." You can think of it as a peer support group for mental health issues in the infosec community. We run villages and events at conferences, hosting things like tables full of fidget toys, adult coloring, and calligraphy, knitting, paracord crafts, tea & snacks, air loungers (for much needed naps), professional massage therapists giving free chair massages, yoga & meditation sessions, quiet music, soft lighting, therapy dogs, etc. We also sometimes have full talk tracks with presentations, discussion groups, and interactive sessions given by both industry peers and medical health professionals

Honestly, it snowballed into the non-profit that it is today (with our amazing group of volunteers and board members) simply because of the overwhelming response after the first "village" we ran in 2018.

Q: WHAT'S THE COOLEST THING YOU'VE SEEN IN YOUR LINE OF WORK?

A: So many things! If we focus on the mental health aspect, I have to say it's the amount of people we've impacted. When I first started, I was blown away by how many people thought they were struggling alone. Whether it was with ADHD, imposter syndrome, thoughts of suicide, depression, and a whole other range of issues — the realization that we all have mental health was just not something people were open about in the community. I've had so many people walk up to me, tell me their story, their struggles, and hug and cry on my shoulder. We're surrounded by so many amazing and supportive people in this community, and it's the coolest thing ever.

Q: WHAT'S YOUR FAVORITE ACTIVITY TO DE-STRESS?

A: Drawing. I only recently really picked it up. I had always doodled on my kid's bag lunches or just randomly for fun. When a friend gifted me some art supplies, I decided to try it for real. It's one of the few things that completely turns my brain off from everything else constantly running through it. I'm still my own harshest critic when it comes to the end result, but practice makes perfect, and I just recently finished my first 200-page sketchbook!

Q: WHAT PIECE OF WISDOM WOULD YOU MOST LIKE TO IMPART TO THE PEOPLE?

A: One of my favorite pieces of advice — and I honestly can't remember where I heard it — is to never take criticism from someone that you wouldn't ask for advice. When my book first hit Amazon, I found myself reading reviews and being incredibly upset by them. But I heard this piece of advice and realized that none of the negative reviews mattered. I didn't know these people, they obviously didn't read the "who this book is for" section, and I have enough kind and brutally honest people in my corner who would have told me anything that I needed to hear.

ADVERSARIAL SIMULATIONS

BY MAX BOEHNER

Do you have detections in place that have never alerted? If so, how can you be sure they work? If you've ever asked yourself this question, keep reading.

Adversarial Simulation (AdvSim) is an activity in which tactics, techniques, and procedures (TTPs) are performed in a controlled manner to simulate behavior of adversaries. In other words: **executing attacker tools and techniques.** There are multiple reasons for running **AdvSim** but for now, we will focus on the aspect of testing detection mechanisms.

To start off, here are some things we will need:

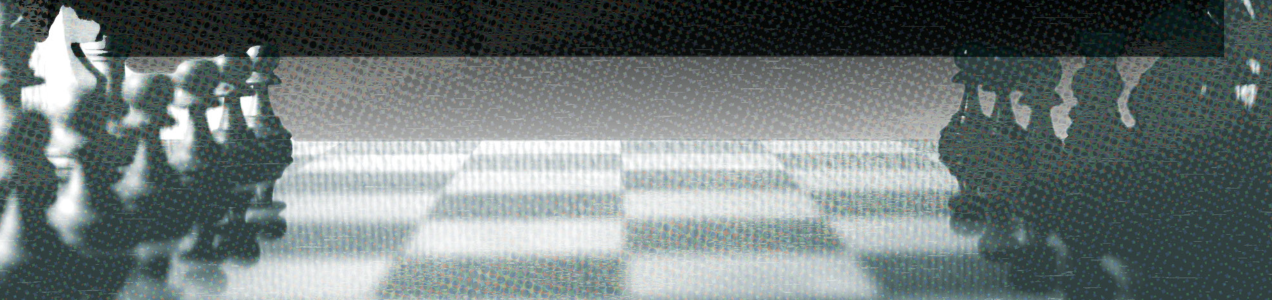
A machine to run our adversarial techniques (AdvSim box)

A user account with local administrator privileges on the AdvSim box

A detection tool, most likely a SIEM (Security Information and Event Management)

The AdvSim box should be representative of a typical system in your environment and there needs to be a mechanism for forwarding its logs into the detection tool [check out our webcast on this subject!]. Using a production machine to run tests is not a good idea, as some tests may jeopardize the integrity and availability of the AdvSim box. **Ideally, a virtual machine is used** that can be reset to a safe snapshot if anything goes wrong (learn how to do this in the next article!). A simple test to ensure that things are set up correctly is to run a program like Notepad on the AdvSim box and then check whether the execution log can be found in your detection tool.

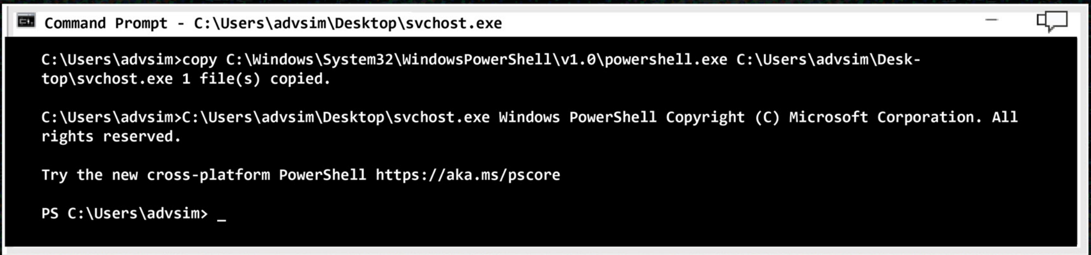
Once we know that logs are flowing, we can move on to picking a detection to test. Assume that we have a detection in place that looks for adversaries launching renamed Windows executables in order to avoid detection. Such a detection can be found in **the Sigma project.**



For the purpose of this test, we will focus on the **PowerShell** executable. If we want to test this detection manually, we could do this by opening a **cmd.exe** terminal and copying the legitimate **powershell.exe** to the desktop of user **Advsim** and naming it **svchost.exe**. Then we simply run that fake **svchost.exe**.

```
C:\Users\advsim> copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe C:\Users\advsim\Desktop\svchost.exe
```

```
C:\Users\advsim> C:\Users\advsim\Desktop\svchost.exe
```



```
C:\Users\advsim>copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe C:\Users\advsim\Desk-
top\svchost.exe 1 file(s) copied.

C:\Users\advsim>C:\Users\advsim\Desktop\svchost.exe Windows PowerShell Copyright (C) Microsoft Corporation. All
rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore

PS C:\Users\advsim> _
```

Now all that is left to do is check our detection to see if it fired. If it did not, you may need to dig into your event logs to determine why. This detection relies on the logs containing something typically called **OriginalFileName** which is an attribute included in **Portable Executables (PE)** at compile time. This information is included in Sysmon Event ID 1 logs but not in built-in Windows logs.

Now that we have tested our detection manually, you might be wondering if there is an easier and more automated way to do this. Luckily, there is a project called **Atomic Red Team** that does just that.

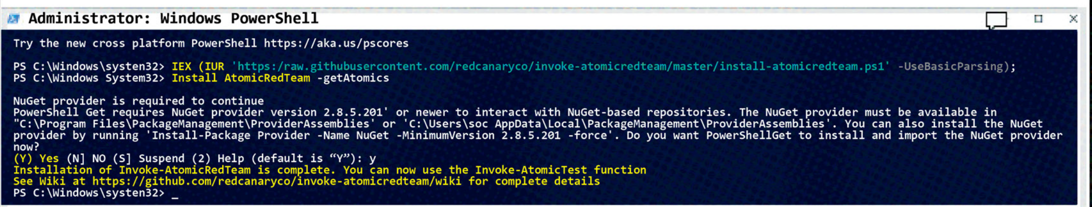
Atomic Red Team is a library of so-called atomic tests intended for testing detections. For extra awesomeness, these tests are mapped to the MITRE ATT&CK® framework. There is an atomic which performs a similar test to the manual one we executed. It is named "Atomic Test #5 - Masquerading - powershell.exe running as taskhostw.exe" and can be found under the category "T1036.003 - Rename System Utilities".

Now, how do we execute this atomic? There are multiple tools out there that can run them. We will be using Invoke-AtomicRedTeam because it is free and easily installed via PowerShell.

To install both **Atomic Red Team** (the atomics) and **Invoke-AtomicRedTeam**, we open a **PowerShell** window as administrator and run the following commands. Please note that you will need to disable any antivirus or EDR solution on the **AdvSim** box to avoid problems. The atomics contain scripts for simulating malicious activity so some of the files will look like malware.

```
IEX (IWR ‘https:/raw.githubusercontent.com/redcanaryco/invokeatomicredteam/master/install-atomicredteam.ps1’ -UseBasicParsing);
Install-AtomicRedTeam -getAtomics
```

If you are prompted about the installation of **NuGet/PowerShellGet**, allow it.



```
Try the new cross platform PowerShell https://aka.us/pscores

PS C:\Windows\systen32> IEX (IUR 'https:/raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/install-atomicredteam.ps1' -UseBasicParsing);
PS C:\Windows System32> Install AtomicRedTeam -getAtomics

NuGet provider is required to continue
PowerShell Get requires NuGet provider version 2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet provider must be available in
"C:\Program Files\PackageManagement\ProviderAssemblies' or 'C:\Users\soc AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet
provider by running 'Install-Package Provider -Name NuGet -MinimumVersion 2.8.5.201 -force'. Do you want PowerShellGet to install and import the NuGet provider
now?
(Y) Yes (N] NO (S] Suspend (2) Help (default is “Y”): y
Installation of Invoke-AtomicRedTeam is complete. You can now use the Invoke-AtomicTest function
See Wiki at https://github.com/redcanaryco/invoke-atomicredteam/wiki for complete details
PS C:\Windows\systen32> _
```

In the same **PowerShell** window, we now have access to the **Invoke-AtomicTest** commandlet. Before executing our atomic, we can look at its details. Remember that we want to execute **test #5** for technique **T1036.003**. To obtain details about the test, run:

```
Invoke-AtomicTest T1036.003 -TestNum 5 -ShowDetails
```

This tells us that the test will **copy powershell.exe into taskhostw.exe and subsequently run** that copied file. This is not exactly what we did earlier but similar, and if we wanted to, we could use custom arguments to specify the target file name to be **svchost.exe**. For the sake of simplicity, we will skip custom arguments. Now run the commands to first execute the atomic and then perform clean-up, which will remove our renamed **PowerShell** file:

```
Invoke-AtomicTest T1036.003 -TestNum 5
Invoke-AtomicTest T1036.003 -TestNum 5 -Cleanup
```

Once again, we will want to verify that our detection fired. If our manual test was detected, then this should typically also have worked.

Armed with this knowledge, we can look at **how to do AdvSim at scale**. One of the benefits of using **Atomic Red Team** over manual testing is that it is easy to create scripts that import and utilize the atomics. This allows us to assemble sets of atomic tests for our detections and execute them periodically to ensure our detections are working as expected. Think of this like unit tests for detections.

Another interesting use case for **Atomic Red Team** is running atomics for techniques relevant for your environment and testing if you already have detections for those in place. You could even assemble chains of attack techniques based on threat intelligence reports and run those. Lastly, if you are very confident about your detection capabilities, you should consider having **AdvSim** performed with the specific goal of trying to evade detection. This can be useful to determine how strong your detections are when faced with advanced adversaries that try to be stealthy.

RESOURCES

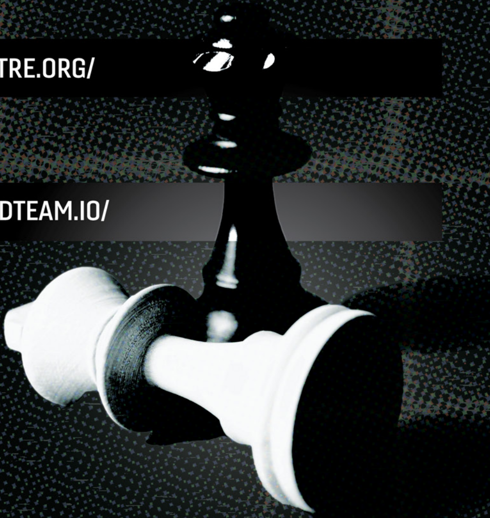
Intro to windows event collecting: https:/www.youtube.com/watch?v=Eix5BPta56E

Sigma: https:/github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_renamed_binary_highly_relevant.yml

Invoke-AtomicRedTeam: https:/github.com/redcanaryco/invoke-atomicredteam

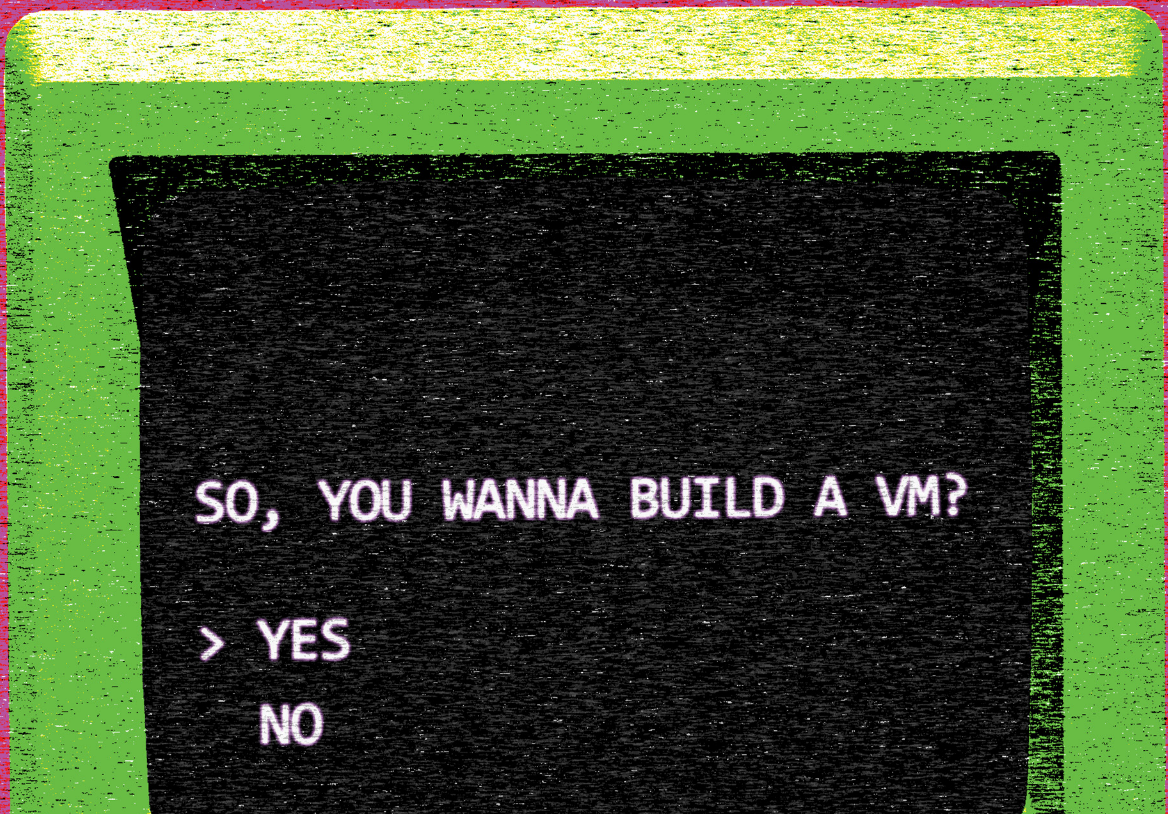
MITRE ATT&CK®: HTTPS:/ATTACK.MITRE.ORG/

ATOMIC RED TEAM: HTTPS:/ATOMICREDTEAM.IO/



CHECK OUT CARRIE'S CLASS:

3 . --- ..-. # ..- ... .. -. -.. .. ...-



SO, YOU WANNA BUILD A VM?

BY WILLIAM STEARNS

PHYSICAL COMPUTERS are easy to understand — I can pick them up, plug cables into them, and carry them around in a backpack.

SO, WHAT'S A VIRTUAL MACHINE?

Imagine taking a magic laser pointer inside the case of a physical computer and drawing lines over the processor, memory, and hard drive, splitting each into four pieces. The original physical computer gets, say, one quarter of the processor cores, memory, and disk space; the other three quarters get divvied up among three virtual machines. These can get started and stopped inside the physical machine. When I run an operating system or program inside one of these, it thinks it's running on a normal computer, when in fact it's only getting a fraction of the original system.

WHY BE ISOLATED?

There's an advantage to breaking the system up this way. We can isolate programs or groups of programs, placing each inside their own virtual machine. I might have my personal web browser and applications in one virtual machine, my development tools in a second, and my security programs, keys, and passwords in a third. If my personal web browser becomes infected, that malware can't reach the other two virtual machines — I've contained the damage.

(If this scenario interests you, see https://www.qubes-os.org/ for an entire laptop setup based on virtualization.)

A virtual machine provides many uses, including the ability to test malware, OS patches, and software installs and upgrades to look for problems. They're also handy for sharing the resources of a single large machine instead of buying lots of smaller ones.

HOW DO I SET UP A VIRTUAL MACHINE?

First, you need a software package like VMWare Workstation, VirtualBox, KVM, or Xen. When choosing, you need to make sure that it will run on your host operating system (the one installed on the physical computer). Some of these are free, some are commercial. Pay attention to the license — some commercial ones are free for personal use.

You'll install this on your host computer. This is the laptop, desktop, or rackmount machine that has some extra memory, CPU power, and disk space.

CREATING THE VIRTUAL MACHINE

With each virtual machine, ask yourself: If I was running this program on a physical computer, how many processors, how much memory, and how much disk space would I need for the program and the operating system underneath? Remember those numbers.

If your virtual machine software package isn't running, start it up. Choose "Create Virtual Machine". If you're offered a virtual machine that has your operating system pre-installed, choose that! It'll save you a lot of time. If you only get the option to create a blank virtual machine, no problem; we'll install the OS in a minute.

As you're creating this, you'll be asked how many processors, how much memory, and how much disk space to give it. Put in the numbers you chose above. Remember that it's generally pretty easy to add or remove memory and/or processors. If you run out of disk space in a virtual machine, most will allow you to add that as well, but keep in mind that removing disk space is not usually possible, so start small and add disk if needed.

You may also be asked if you want to 1) pre-allocate all disk space or 2) allocate on demand. Choice 1 means that if you want a virtual machine with 200GB of disk space, you will immediately tie up 200GB of disk space on your host. If you allocate on demand, you'll only use a small portion of that — perhaps 10GB-20GB for the initial OS install, but that number will grow as you add more files inside the virtual machine. The latter is probably a good call for most people.

Finally, when you get to networking options, you'll very likely pick something like "allow outbound networking". The main exception to this is when you're testing malware and want to totally prevent the malware from making network connections.

There will be a ton of other options, and in most cases, you can ignore all of them.

LOADING THE OS

If you weren't offered a pre-installed operating system, you'll have to install one. Here's where you'll need either a DVD or an ISO file downloaded from your operating system vendor (place this somewhere you can find it on your host hard drive). Go back to your virtual machine configuration, look for disks/drives, and in particular, the virtual DVD drive. You'll have the option of pointing this at your host's DVD drive or a file on the host's hard drive! Here's where you select the ISO file you downloaded. When you first start the virtual machine, there won't be an operating system, so the VM will try to boot from what it believes is the DVD drive (but is actually that file on the host!). This will let you do the operating system install.

When the OS is installed, go back to the VM configuration and detach this ISO file from the virtual DVD drive and restart the system. You should boot up into the new virtual machine.

STARTING, STOPPING, AND SNAPPING!

Once you've pulled down and installed all OS patches, shut down the system. Go back to your virtual machine software on the host, select this new virtual machine, and create a snapshot. This is a copy of the virtual machine that is also stored on your host hard drive. Give it a name like "Ubuntu Linux 20.04 patched 20220711" so you know what's in it.

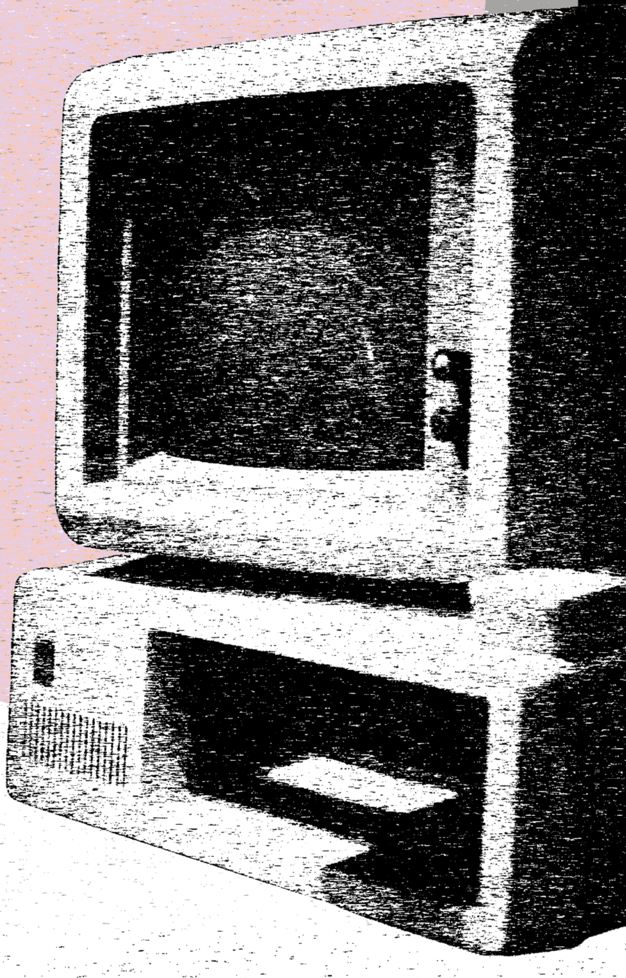
If something goes wrong with the virtual machine — it gets corrupted, infected, or you just plain want to start your testing again — you can stop the VM, revert to an old snapshot, and be right back to where you were without having to start over at ground zero! When I'm working with these, I'll usually make a snapshot right after I've installed the OS and applied all patches, another after I've made all my custom modifications (like accounts, keys, additional support packages, etc.), and one final snapshot after installing the software I'm testing — but before I start it up for the first time. This means I can revert to any of those stages quite easily.

There's one final advantage to taking snapshots. When it comes time to create a brand-new virtual machine for a new project, you can use one of your existing snapshots as a starting point! This means you get to skip all the steps you had to take to get there in the first place.

SO, WHAT DID WE GET?

With a physical computer, we have a mindset that it is a resource to be protected. We don't want anything to corrupt the drive, screw up the configuration, or otherwise make it unavailable.

With a virtual machine, we get the exact opposite — you should get used to the idea that you can create and delete these at any time! If something goes wrong, just go back to a previous snapshot and try again.



HI THIS IS MY ARTICLE I AM SORRY IT'S WAY LONG I AM TOO NOOB SO MUCH FLUFF KASSIE PLEASE FIX IT OH BOY HERE WE GO

JOHN, DON'T FIRE ME

By Some SOC Analyst

A.K.A. Cyber Deception by Kaitlyn Wimberley

Deception as a technique in physical confrontation is not a new concept. Think Trojan horse or trapdoors. Think misinformation and psychological warfare. Think Mark Rober's glitter bombs. Everyone goes to Sun Tzu for war advice, and he said, "All warfare is based on deception."

We all know about deception used by attackers — phishing emails, social engineering, sketchy executables disguised as something benign, etc. But why should the offense have all the fun? Why would the blue team build up our defenses, and then sit around waiting to be attacked? It's good for us to build our moats and walls, but what about the guys (and gals) that sneak in? (And they will.)

Any well-rounded defensive program should include cyber deception. But what is it? Why should you consider it? Is it difficult to implement? (Hint: it's not.)

What is Cyber Deception?

Cyber deception is a form of active defense. It often involves setting up a juicy target that is rigged to help detect, identify, distract, or deter attackers (or Annoy, Attribute, and Attack... because apparently AAA is a cool acronym that is underused). What are the benefits?

It's obvious that detection is important. You want to know when you are actively being targeted. Deception can be used to fill in detective gaps. Maybe the execution of that tool was missed, but a canary account was touched and it set off the alarm. The more lasers the bad guy has to avoid, the better.

We also have a great opportunity to use deception to find out WHO is attacking us. We can follow the trail from that tripped canary to see what host has been compromised (or maybe it's an insider threat?). A honeypot can be monitored to see who is trying to take advantage of it. Poisoned documents can serve as our own Trojan horse that calls home with the information of whoever stole it. If we know who is attacking us, we may learn why they are targeting us, what techniques they use, or how skilled they are.

Another hugely important benefit of deception is wasting the attackers' time. We get to be really annoying and creative here. We point them to empty chests, and they break them open to find no treasure. We can poison their enumeration, or slow down their attack with a tarpit. The more times a hacker fails and goes after another target, the more chances we have to detect them. The more time spent wasted, the less time they spend succeeding.

Of course, the ultimate win with deception is deterring an attacker from continuing their attack or going after us at all. Annoyance can tie into this. A fly-by attacker isn't going to waste their time in an obnoxious environment when they just want an easy win. Not everyone will be deterred, but in general, attackers want to take the path of least resistance.

Making Deception Work for You

Effective implementation requires action. It does no good to incorporate deception if it is not being monitored or nothing is done with the information. Create detections for canary accounts (and tune them!), quarantine devices, block the attackers out, use the knowledge gained to inform defensive decisions.

Cyber deception is also not magic. It cannot, and should not, replace the fundamentals (patching systems, firewalls, segmentation, security policies, etc.). Deception should be the cilantro on the taco (icing on the cake is overdone) for your defensive strategy. It should fill gaps and provide visibility and intelligence where there was less or none previously.

Finally, always operate with integrity. Cyber deception is fun, but keep in mind the needs and policies of your organization and the current limits of the law. We're not here to be vigilantes or law enforcement (unless you actually are). Protect what belongs to you, but don't become the bad guy to get back at one.

Are We Done Yet?

Almost.

There are so many wickedly brilliant deception tactics and tools out there. There are tools and techniques you can use to not only fool the attackers, but also attribute attacks to them, or even attack back (legally). If you're at all interested in deception and learning some (only a little evil) tricks, check out John's class, Active Defense and Cyber Deception. This article is a pond; John's course is the ocean.

Now, let's go set some trip wires and make the attackers catch themselves!

RESOURCES

John's Class: https://www.antisyphontraining.com/active-defense-cyber-deception-w-john-strand/

ADHD: https://adhdproject.github.io/

Glitter Bombs: https://www.youtube.com/watch?v=xoxhDk-hwuo

CHECK OUT A FEW
EXAMPLES AND
RESOURCES ON THE
NEXT PAGES TO GET YOU
STARTED AND
WHET YOUR APPETITE.

CHECK, CHECK, CHECK, CHECK, CHECK IT OUT!

KAITLYN WIMBERLEY

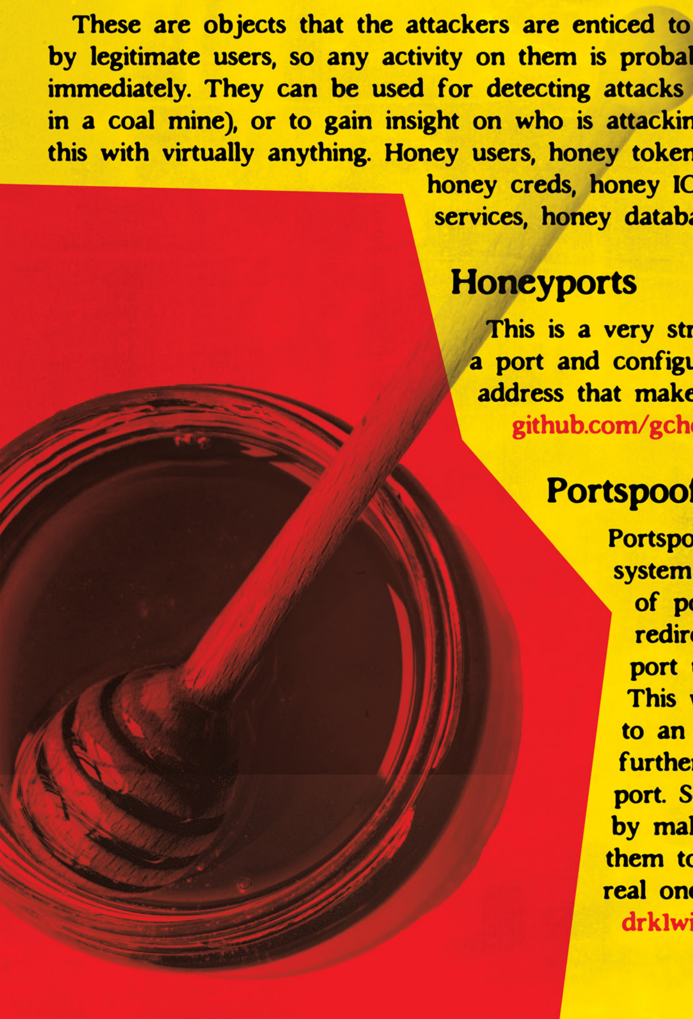
ADHD

The ADHD Project is a resource that has a collection of deceptive tools divided into the three categories of Annoyance, Attribution, and Attack. This is a great place to start looking at all of the gloriously mischievous things you can do with deception.

Adhdproject.github.io

Honeypots

These are objects that the attackers are enticed to interact with. They are never touched by legitimate users, so any activity on them is probably malicious and should be investigated immediately. They can be used for detecting attacks that other defenses miss (like a canary in a coal mine), or to gain insight on who is attacking you, and how, and why. You can do this with virtually anything. Honey users, honey tokens, honey ports, honey IOT, honey docs, honey creds, honey ICS, honey networks, honey accounts, honey services, honey databases... the list goes on and on.



Honeyports

This is a very straightforward python script that listens on a port and configures the firewall to block any source IP address that makes a connection to that port.

github.com/gchetrick/honeyports

Portspoof

Portspoof camouflages services running on a system and slows down an attacker's enumeration of ports and services. It accomplishes this by redirecting any packets received on a TCP port to the port that Portspoof is listening on. This will make it appear that all ports are open to an nmap scan. You can even take it a step further and spoof service signatures on each port. So, the attacker has been slowed down both by making scans take longer and by forcing them to parse through feigned services to find the real ones.

drk1wi.github.io/portspoof/

Canary Accounts

Setting up a canary account can take less than five minutes. Do it. It can be as simple as creating a user with a very strong password and strong login restrictions (disable login hours, etc.). If you detect someone trying to log in to this account, it may be a password spray or some other sketchy activity. This is really easy to implement and could even supply some basic actionable monitoring in an environment while working on those fundamentals mentioned earlier. Check out John's article, Canary Accounts in Active Directory, in PROMPT#'s Better Together issue.

Traps and Tricks

A few more fun ways to mess with attackers.

Spidertrap

Spidertrap tangles up web crawlers into an infinite set of generated web pages. The webpages are generated on the spot and contain links that lead to another page full of links that lead to another page full of links that lead to another page full of links... you get the idea. This continues until either the crawler is stopped, or the script is.

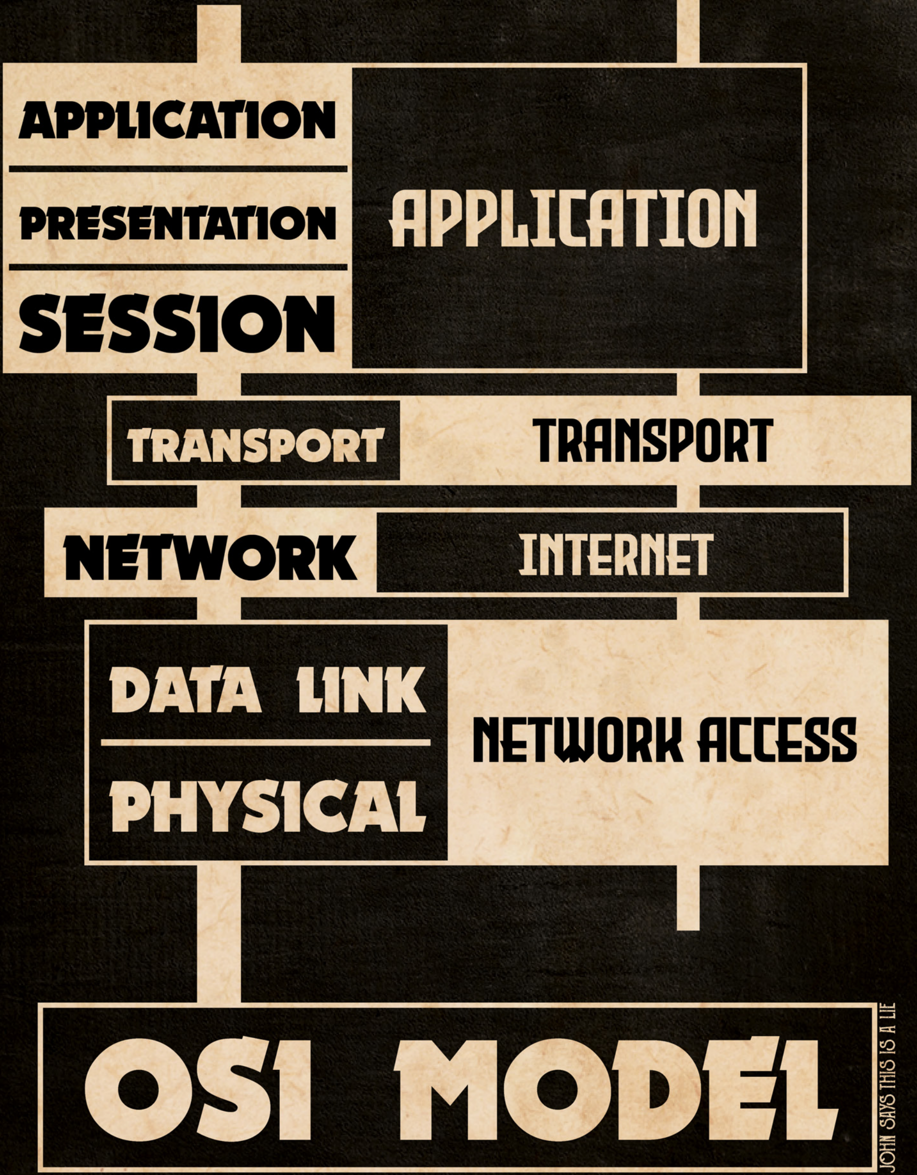
github.com/adhdproject/spidertrap

PHP-HTTP-Tarpit

The tarpit has several ways of dealing with bots and web scanners. There are multiple modes for dealing with requests, or you can just rotate through the modes randomly. The tool can respond to all requests with Success 200 responses filled with loaded random garbage content to trigger false positives. It can redirect all requests back to the source IP to waste time and potentially the attacker's resources. Or, the namesake Tarpit mode will return status codes like 101 or 104, and steadily send further responses more and more slowly. There are other equally frustrating modes as well.

github.com/msigley/PHP-HTTP-Tarpit

TCP/IP



OSI MODEL	TCP/IP
APPLICATION PRESENTATION SESSION	APPLICATION
TRANSPORT	TRANSPORT
NETWORK	INTERNET
DATA LINK PHYSICAL	NETWORK ACCESS

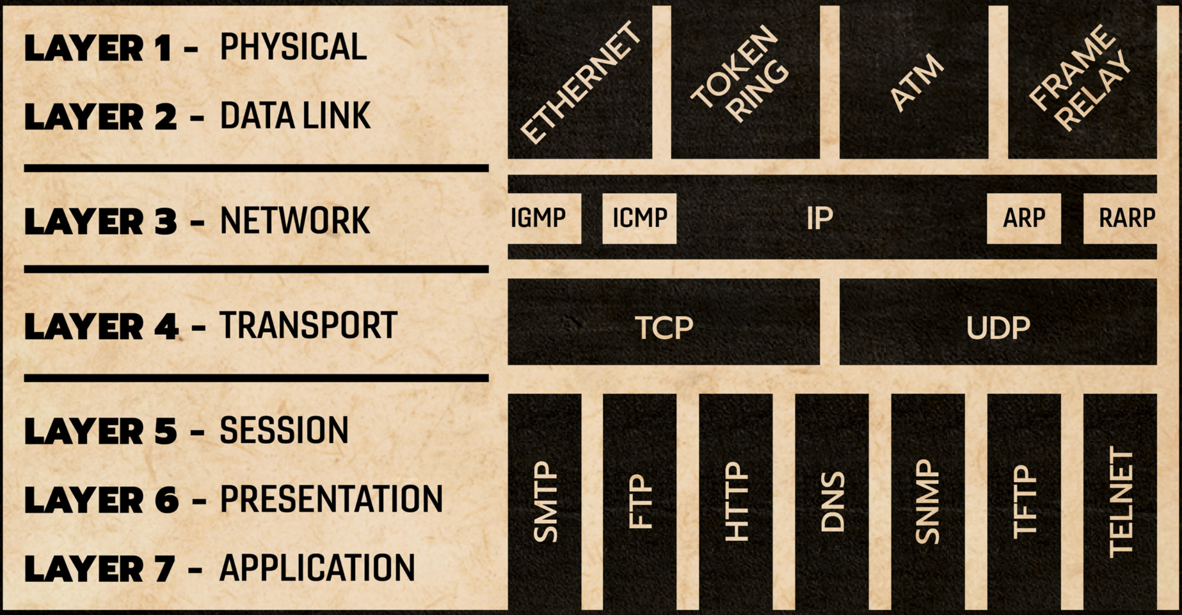
OSI MODEL

JOHN SAYS THIS IS A LIE

BLACK HILLS
Information Security

SIDE A

PROTOCOLS:



LAYER 1 - PHYSICAL

LAYER 2 - DATA LINK

ETHERNET | TOKEN RING | ATM | FRAME RELAY

LAYER 3 - NETWORK

IGMP | ICMP | IP | ARP | RARP

LAYER 4 - TRANSPORT

TCP | UDP

LAYER 5 - SESSION

LAYER 6 - PRESENTATION

LAYER 7 - APPLICATION

SMTP | FTP | HTTP | DNS | SNMP | TFTP | TELNET

SIDE B

NETWORK ACCESS

INTERNET

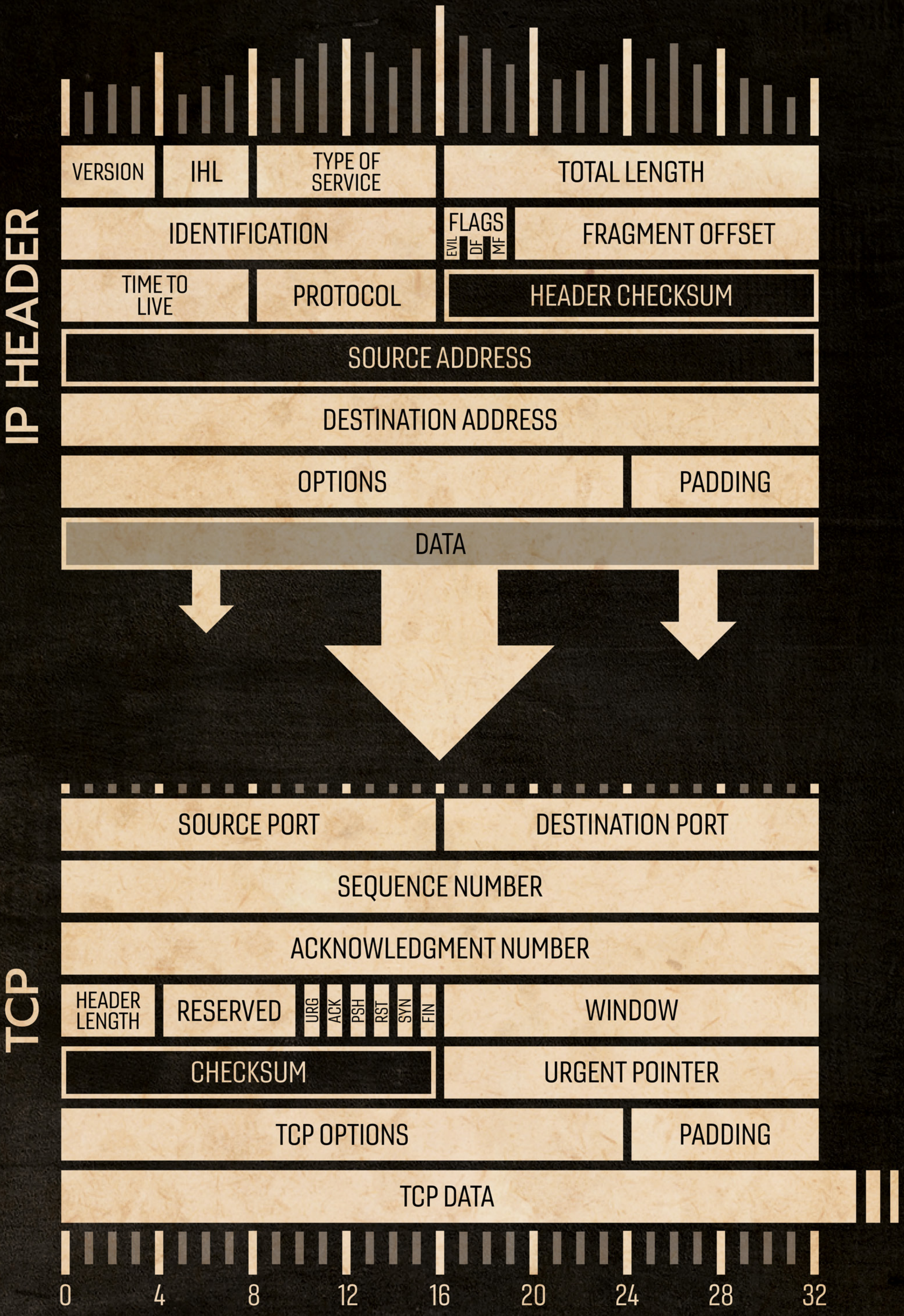
TRANSPORT

APPLICATION

This is the internet. Know it. Love it. Breathe it. It's the underlying structure to how all computers connect. From cables and handshakes to ports and protocols, these standards make the internet useable for kind souls and black hats.

Use it wisely. Use it well.

BLACK HILLS
Information Security



IP HEADER

VERSION | IHL | TYPE OF SERVICE | TOTAL LENGTH

IDENTIFICATION | FLAGS (EVIL, DF, MF) | FRAGMENT OFFSET

TIME TO LIVE | PROTOCOL | HEADER CHECKSUM

SOURCE ADDRESS

DESTINATION ADDRESS

OPTIONS | PADDING

DATA

TCP

SOURCE PORT | DESTINATION PORT

SEQUENCE NUMBER

ACKNOWLEDGMENT NUMBER

HEADER LENGTH | RESERVED | URG | ACK | PSH | RST | SYN | FIN | WINDOW

CHECKSUM | URGENT POINTER

TCP OPTIONS | PADDING

TCP DATA

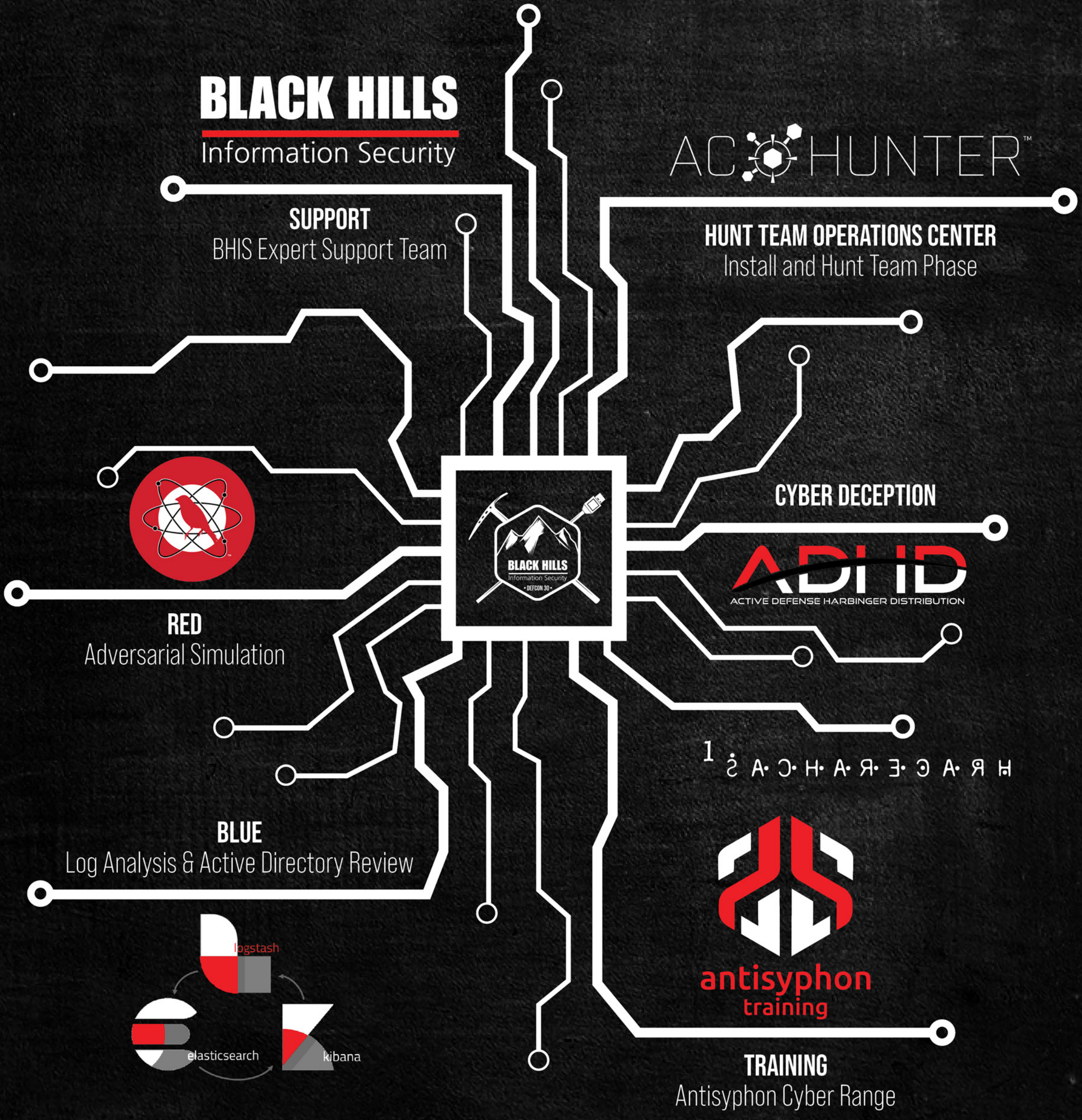
0 4 8 12 16 20 24 28 32

ANATOMY OF A PACKET

BHIS Active SOC

PENETRATION TESTING is where everything started for us. We're ***hackers*** — we know the things that **DON'T** stop attackers… and the things that ***DO.*** So, we developed an Active SOC approach to securing our customers.

CHECK IT OUT:



SUPPORT
BHIS Expert Support Team

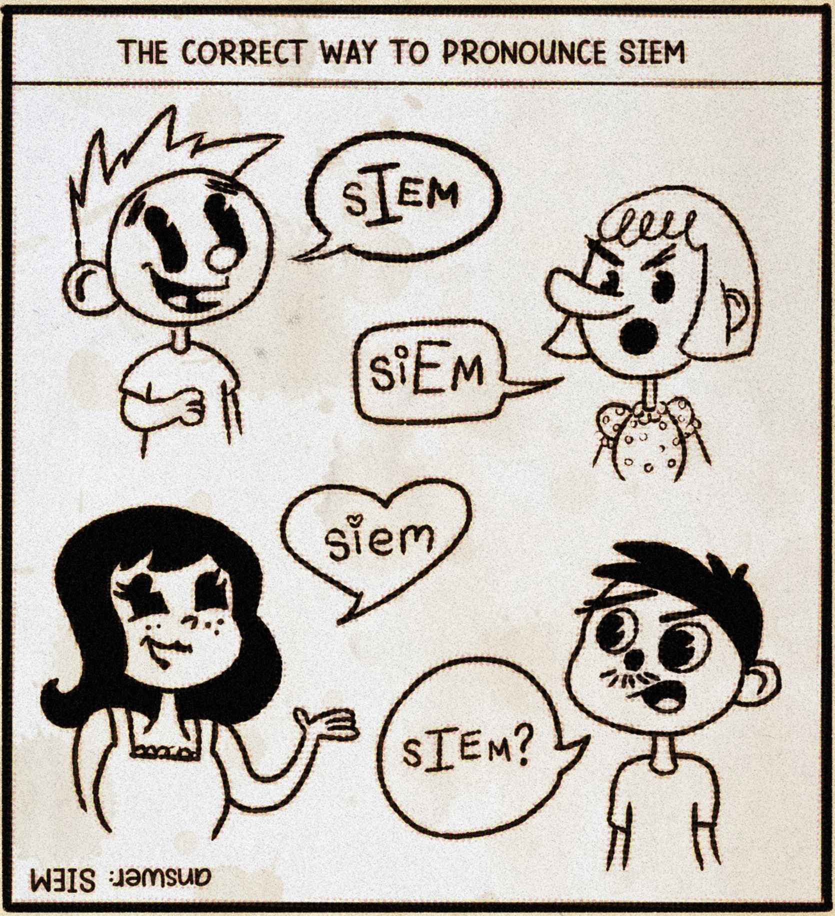
HUNT TEAM OPERATIONS CENTER
Install and Hunt Team Phase

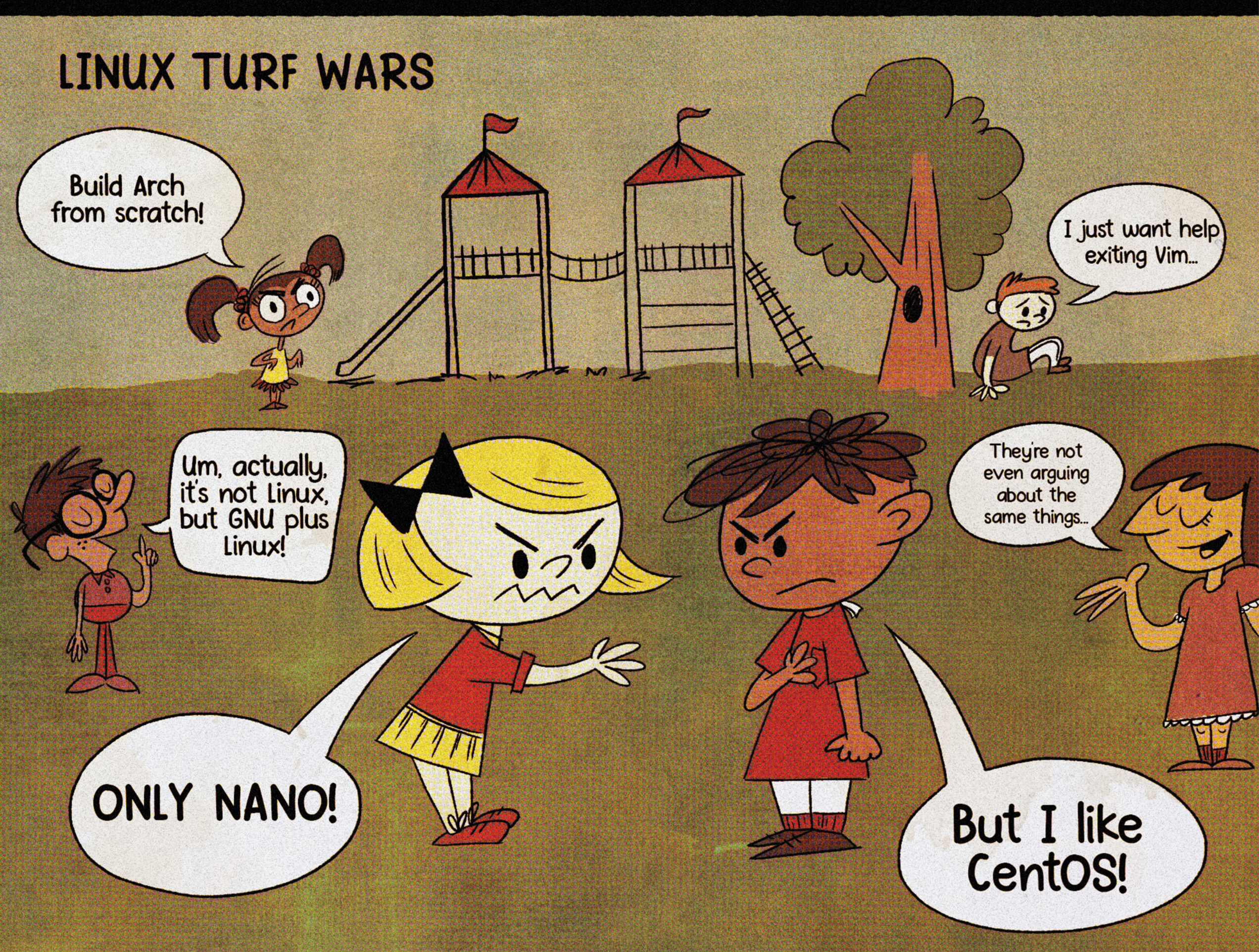
RED
Adversarial Simulation

CYBER DECEPTION

BLUE
Log Analysis & Active Directory Review

TRAINING
Antisyphon Cyber Range


THE CORRECT WAY TO PRONOUNCE SIEM
sIEM
siEM
siem
sIEM?
answer: SIEM

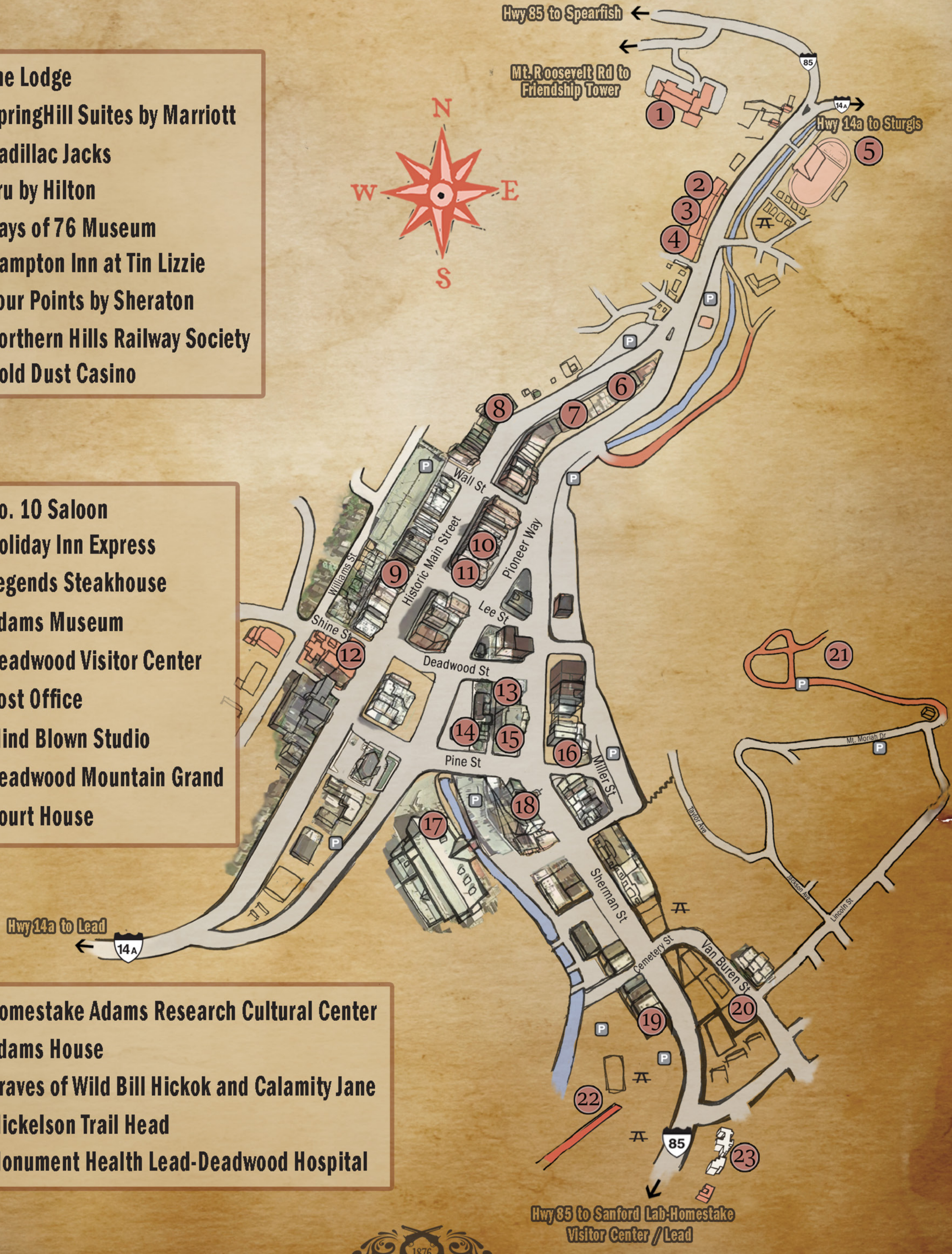

LINUX TURF WARS
Build Arch from scratch!
I just want help exiting Vim...
Um, actually, it's not Linux, but GNU plus Linux!
They're not even arguing about the same things...
ONLY NANO!
But I like CentOS!

WILD WEST HACKIN' FEST

1. The Lodge
2. SpringHill Suites by Marriott
3. Cadillac Jacks
4. Tru by Hilton
5. Days of 76 Museum
6. Hampton Inn at Tin Lizzie
7. Four Points by Sheraton
8. Northern Hills Railway Society
9. Gold Dust Casino

10. No. 10 Saloon
11. Holiday Inn Express
12. Legends Steakhouse
13. Adams Museum
14. Deadwood Visitor Center
15. Post Office
16. Mind Blown Studio
17. Deadwood Mountain Grand
18. Court House

19. Homestake Adams Research Cultural Center
20. Adams House
21. Graves of Wild Bill Hickok and Calamity Jane
22. Mickelson Trail Head
23. Monument Health Lead-Deadwood Hospital



1876
DEADWOOD

ALERT
SECURITY OPERATIONS CENTER
PROMPT#