

The banner features a dark red background with a subtle pattern of binary code (0s and 1s). A large, stylized red target symbol is positioned on the left side. The main title is in white, bold, sans-serif font. Below the title is the website URL in red. At the bottom left, there is information about the event being live online and the dates for the summit and training. At the bottom right, the SANS DFIR logo is displayed in white.

# Threat Hunting & Incident Response

## Summit & Training

[sans.org/ThreatHunting](https://sans.org/ThreatHunting)

Live Online 

FREE SUMMIT: October 7-8  
TRAINING: October 11-16

**SANS DFIR**

**Common Misconceptions  
and Mistakes Made in  
Threat Hunting**

Christopher Witter, Spotify

Live Online 

**FREE SUMMIT:** October 7–8  
**TRAINING:** October 11–16

 **SANS**  
**Threat Hunting**  
Summit & Training

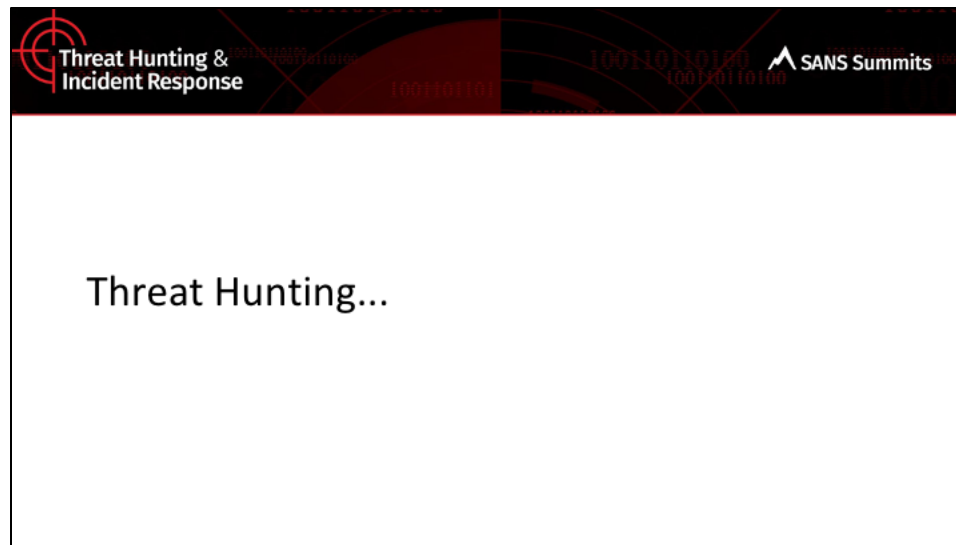
The banner features a dark red background with binary code (0s and 1s) scattered across it. On the right side, there is a circular inset photo of Christopher Witter, a man with a beard and sunglasses, wearing a blue jacket. The text is white and black, providing high contrast against the background.

Threat hunting is the act of finding or uncovering previously identified intrusions or TTPs via means outside of the standard SOC operations processes. We are performing these actions to MINIMIZE DWELL TIME, and discover new ways of detecting intruders and their activities.

About me:

- Founding team member Falcon Overwatch
- Founding team member DoD contractor CSIRT
- (3) DFIR Summit Speaker
- Spotify, Detection and Response
- DFIR Nerd
- Outdoor Enthusiast
- Maker (3D printing, CNC Machining)





Threat hunting is the act of finding\uncovering previously unidentified intrusions or TTPs via means outside of the standard SOC operations processes. We are performing these actions to MINIMIZE DWELL TIME, and discover new ways of detecting intruders and their activities.



Threat Hunting & Incident Response

Misconception


SANS Summits

The Baseline Fallacy...

The slide features a dark header with a red target icon on the left, the text 'Threat Hunting & Incident Response' in white, the word 'Misconception' in large white font in the center, and the 'SANS Summits' logo on the right. The background of the header is decorated with faint binary code (0s and 1s). The main content area is white with a black border and contains the text 'The Baseline Fallacy...' in black.

A Baseline is establishing what is good or normal within your environment.

This fallacy is thinking you need to establish a baseline of your environment in order to perform Threat Hunting. Baselineing can be a time consuming process and instead of being a separate event or process it should be considered built in. But how...



**If not a baseline then what?**

“Once is happenstance. Twice is coincidence.  
Three times is enemy action” - Ian Fleming

Enter -> The Rule of **3's**

If I don't need a baseline, what do I need?

Data... Enough data to establish a pattern of activity or to validate the activity as normal. While wanting to know what's GOOD, how do you know whatever it is you're looking at isn't already COMPROMISED...You don't necessarily until you've investigated it. When it comes to creating or establishing a pattern I like this quote:

“Once is happenstance. Twice is coincidence. Three times is enemy action” - Ian Fleming, James Bond: Goldfinger

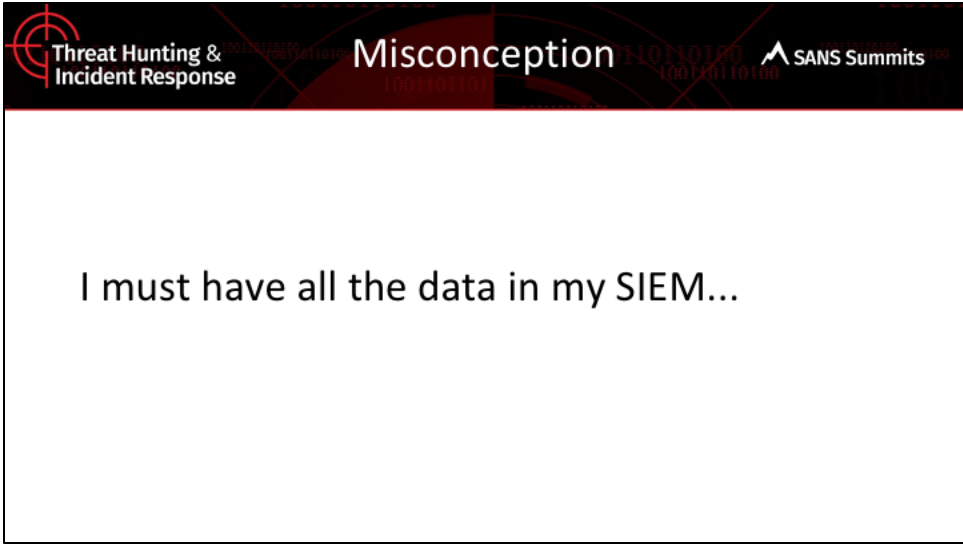
While three times may not necessarily be ENEMY action it is certainly a PATTERN at this point wither it's the users or the bad guys.

When it comes to establishing or identifying a pattern I like to use the RULE of 3's

The rule of 3s: 3 days, 3 weeks, 3 months, 3 fiscal quarters etc.

Looks for things that happen 3 or more times: daily, weekly, monthly...Or even 3 times at all, an outlier in your environment is exactly that an outlier and may not be related to normal operations. If you only have enough data to SEE IT once there's no real way to know if it's normal or not.

Ex: Imagine you find an Office application calling CMD.exe and EXECUTING FTP.  
Super sketchy from the beginning.  
Now imagine you've determined the user is in accounting... That's a bad sign, a non technical user with that sort of activity should have your SPIDEY SENSE tingling...  
You search and see that this has happened 2 other times, all at the end of the month...Then you realize this occurrence is on the last day of the month as well...  
Upon contacting the user you find out this is a standard PAYROLL process...that happens monthly :)



Threat Hunting & Incident Response

Misconception

SANS Summits

1001101100

1001101100

1001101100

I must have all the data in my SIEM...

Data volumes are ever increasing along with the number of software and services that organizations are using all the time. While it is nice to have all the data in your SIEM not all SIEMs are created equal in this area. When it comes to being able to actually execute Threat Hunting where data JOINS and log volumes can complicate matters even the best environments can be crushed under the weight of the computational intensive tasks coupled with massive data volumes.

1st) Consider leveraging your existing tools or applications capabilities to narrow the scope OR power your hunts. While this “grow your own method” can be slightly more time consuming and requires more knowledge of the platforms and maybe even some scripting skill it can be used to one’s advantage. Image querying you cloud platform or on-premise systems APIs to execute queries and then reingest that data into another separate tool for analysis, OR even only ingesting that subset into the SIEM. Depending on the approach this process could make JOINS a manual or scripted process at best, but is still doable.

Ex: I had a client running an EDR solution on a couple hundred thousand endpoints that sent it’s logs back to central servers which provided its own search interface. While you could perform searches against the telemetry you couldn’t write your own detections ;( The vendor did however expose an API for the search interface where we could script up searches which equated to “detection rules” we then piped those results into an Elasticsearch



instance for analysis. We immediately started finding things that were evading the EDR's own behavioral and AV detections. All this WINNING with nothing more than an EXISTING TOOL, some python scripting, and a VM. The lesson here is maybe you actually don't need any new tools but you can cobble something operational with things you already have!

2nd) Tools like Velociraptor and OSQuery allow you to query locally stored information with little infrastructure or storage requirements by querying locally stored data and information to perform your Threat Hunting on endpoints. The concept of scanning locally to find Evil is not a new one, Mandiant started with their MIR agent and then later on other vendors and platforms took similar approaches or extended the concept to include collecting information and storing it locally for querying later, EndGame (Elasticsearch Endpoint). If you do decide to implement this method definitely expand the amount of locally stored logs as it's basically free given today's disk sizes as well.

- F-Secure's Countercept team's Chainsaw is definitely worth looking at <https://github.com/countercept/chainsaw>

3rd.) Lastly, there's something often referred to in the professional services industry as a "Compromise Assessment". This typically is where a vendor will come in and deploy some sort of EDR, GRR (Google Rapid Response), or even custom software. They may review your existing SIEM logs and alerts using their own threat intelligence. Lastly, the good ones at least, will collect targeted "Dead Box" artifacts (MFT, Shim Cache, App Cache, registry, Event Logs, etc.) for analysis in mass instead of collecting entire disk images. This method of analysis is very similar to a DFIR technique originally championed by Chris Pogue in his "Sniper Forensics" series of talks.

Threat Hunting & Incident Response

Misconception

SANS Summits

I need an EDR...

EDR vendors would lead you to believe that you need EDR to perform Threat Hunting. Do you? No! Does EDR make Threat Hunting 1000% times easier, YES. Why? Because EDR data may already contain several types of data: IP connections, Domain names, File hashes, Process Command Lines, etc... Not to mention Process ancestry/linkage which makes it WAAAYYY EASIER answering: who, what, when, where, why, and how.

When I started in DFIR at first I only had access to NETWORK telemetry (Proxy, IDS, Full Packet Capture (FPC)), and I used to say if only I had the DISK I could tell you what happened via forensic timeline, after awhile MEMORY forensics came along and was able to fill in even more gaps! When EDR came along it was the perfect mix of information from all THREE separate sources. Immediately I was able to work through and close ~80% of the cases that came along faster and easier without the need for MEMORY or DISK forensics.

All the data you need is in one place, no massive or computationally intensive JOINS of disparate data sets.

So NO, you do NOT need an EDR solution however you do NEED data and potentially LOTS of DATA from different sources.



Threat Hunting & Incident Response

Contextual Sources

SANS Summits

The **further away** from the user you get the **less context** you have and the more data sources you'll need for context \ intent. Some data sources have higher contextual value.

1. Shellcode
2. OS \ Kernel level events
3. Application logs
4. Network Level Information (Firewall, Netflow, etc.)

Since we know we ultimately can't trust a user or their memory we need to RELY on our EVIDENCE to help us assess what the INTENT of the user was or what the context of the activity we're investigating and why.

Example A: Imagine you get a hit for malicious C2 DOMAIN OR IP you just learned about and blacklisted. During a retroactive hunt for that indicator you find there was a host communicating with that indicator days before. By itself that hit (maybe from a web proxy, Firewall, or DNS server) has very LITTLE CONTEXT. A system looked it up or maybe resolved it and that's all you know.

Answer A.) This NETWORK level communication has LITTLE CONTEXT and requires additional dataset JOINS to further add CONTEXT to CONTINUE the investigation. Imagine then you were able to take the SRC IP and tie it back to a specific BU firewall and then obtain the DHCP records to ID that HOST on that specific day. Going even further you can track that HOSTs owner or who was logged in on that day...

Here we have roughly 4 different datasets to get to our answer (Original Data HIT, BU F/W, DHCP, ASSET DB or LOGON). We got there but maybe in a time consuming and elaborate manner! REGARDLESS, IT CAN BE DONE!

Answer B.) EDR would have taken you potentially right down to the URL and host with one query and displayed it all in a pretty UI with a Process Graph. Full of RICH CONTEXT around the process ancestry which would

immediately help identify the INTENT\CONTEXT around the connection, hopefully immediately leading to you being able to close the case OR WORSE start your IR process. It's nice to have this decision and not be forced into an IR to simply determine the nature and validate this communications.

Contextual data sources for hunting RICH to POOR:

1. Shellcode - Ex: Malware Static analysis describing behavioral functions and indicators. It can contain a ton of data\information and context around the who, what, when, where, why, and how the malware operates. These findings can be used to not only further cases but be used in future hunts or building hunts off of, considering you can learn the behavioral actions something takes and not just it's statically configured indicators.
2. OS \Kernel Events - Ex: Auditd, Sysmon, ETW, Event logs, EDR, etc. These data sources alone or in concert with one another depending on the OS can paint pretty complete pictures of INTENT\CONTEXT and in some cases depending on how robust they are will get you 80% of the way.
3. Application Logs - Ex: Duo\Okta, Web Servers, SAAS applications etc. These can be some of the most challenging logs to use and are the most prone to configuration errors or are NON-EXISTENT. If you happened to get access to them and they're configure properly they can lead you to understand the changes and actions that took place on the platform in a step-by-step manner. Don't expect them to be configured properly unless you've engaged with the teams prior to your hunting and security adventures.
4. Network Level - Ex: Firewall, DNS, Proxy, Netflow etc. These logs can have varying degrees of detail and CONTEXTUAL value. Sometimes they are required to get down to the final enduser\endpoint and sometimes they provide robust attack information, it all depends. Given they typically lack a lot of CONTEXTUAL value you'll more then likely need multiple different ones to fill in the blanks.

Threat Hunting & Incident Response

Mistake

SANS Summits

Only applying Hypothesis driven hunting to standard data sources in a routine manner...

Using the Scientific Method OR Hypothesis driven hunting exclusive to vendor specific datasets and telemetry is one of the biggest mistakes I've seen in the past.

Ex: Actor group XYZ uses Powershell scripts to infiltrate systems using XYZ PS code or commands.

Hypothesis driven hunting is often over generalized. The example above should be covered by retroactively searching your logs\hosts for the specific behaviors or indicators you're interested in, which is typical how it's applied in majority of cases I've run into.

In the past I've rarely seen Hypothesis driven hunting being leverage on the CROWN JEWELS... Every environment typically has something that is entirely UNIQUE with in their environment customer built, bespoke, stove pipe systems... Think about what drives your business: ticketing systems, seating, data processing systems, custom IAM systems, etc. Depending on your industry vertical it could be anything. One think you'll probably come to learn is most of these systems certainly shouldn't be connected to the Internet let alone connected to a business network :) Ok, maybe that's a bit too far.

Hypothesize... How could these systems be abused? What valuable information do they contain? Chances are those systems identified are CORE to your businesses existencesuccess in some way.

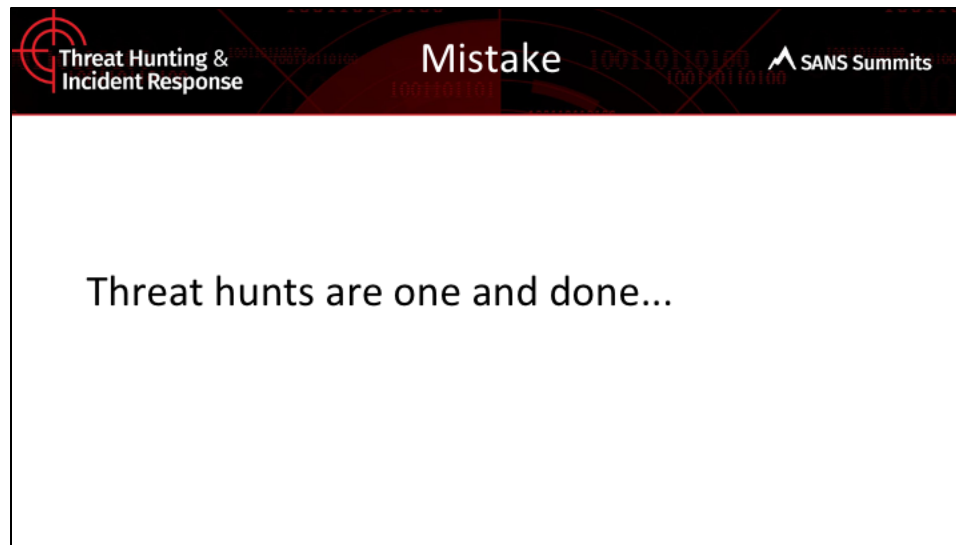
How do we tackle a Hypothesis based hunt here:

- Do I have API \UI logs?
- Do I have auth\access logs?
- Does it record user\admin actions?
- The list goes on!

If you're application is custom built there's a high probability that some if not all of those logs will be missing! Great...Now what?

You'll need to circle back to the CONTEXTUAL DATA SOURCES for answers while you wait for developers to create the logs you need OR laugh in your face because that application is 20yrs old and should have never been web enabled...yet here you are :)

Outside of using the Network level logs to establish communications patterns and apply statistical analysis against them another option would be to try and get Full Packet Captures (FPC) for the host and if it's over HTTPS get the required information to decrypt the traffic for better analysis of traffic etc. It's not much but in the end it's certainly better than nothing and you may at least have a baseline of normal for this precious asset class.



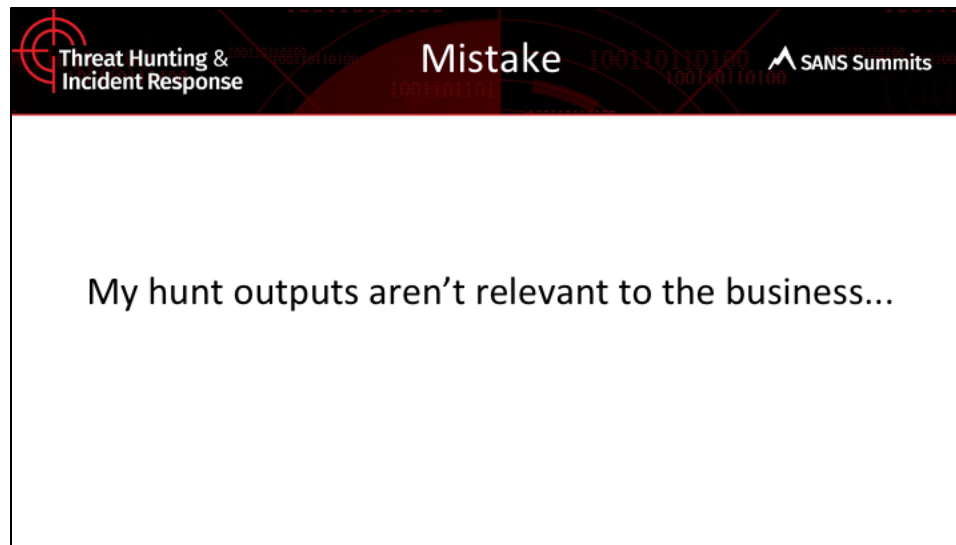
All too often I've talked with customers or seen service providers who would perform a Threat Hunt once and call it GOOD! We did the hunt found X or nothing and now we're done with it and we'll move on to the next hunt. Nothing to see here, next....

The problem with the mentality is you've developed a method for uncovering something you think you would indeed be malicious. If you don't find that badness today there is nothing to say it won't happen tomorrow... is there? After a hunt is complete one of the main things that should be captured is how do we IMPLEMENT this as a permanent DETECTION ??? If it can't be implemented as a detection you should seriously be looking to implement these steps, or DETECTION METHOD as I like to call it, regularly. Regularly can be at a given interval *X* days, once a week, once a month, whatever your organization's temperature is for DWELL TIME. After all we are Threat Hunting to REDUCE the DWELL TIME of a latent intrusion being in our environment. In some organizations, you might fight tooth and nail to get hours to perform your HUNTS. Other orgs there might be a dedicated team, either way if it isn't becoming a dedicated DETECTION you should create a cadence to apply your DETECTION METHODS at REGULAR INTERVALS to ensure that hunt gets EXECUTED..

Scripting and automations are key here, reduce the body of work as much as possible to present the team with only the data that requires human analysis and investigations. If the entire process was automatable these wouldn't be



Threat Hunts they'd be standard detections that the SOC would just investigate.



I had a lot of trouble capturing this one with a one-liner! Honestly, here's the deal... I've yet to see decent reporting for Threat Hunting from a service provider, customer, or internal teams. You're asking for TIME or MONEY or BOTH in order to go off and perform an activity that you hope yields ZERO RESULTS. Are you going to turn in a blank report after hours, days, or weeks? I'd hope not but all too often the output I've seen has been extremely underwhelming if there even were any outputs outside of a short conversation or some basic documentation.

**YOU NEED TO BE SELLING THIS AND CAPTURING THE VALUE IT BRINGS TO YOUR ORGANIZATION!**

Threat Hunting is literally black magic buzzwords CISOs use to sell your efforts internally, but are you actually accurately capturing your body of work in a format that your management can grasp and understand? Can you break down the fruit of your labor into something management can digest? Is your service provider charging you for a service but not capturing and reporting how those "consulting hours" are being used and how these services are actually increasing your detections strengths and your overall security posture... Probably not!

Threat Hunting & Incident Response

Hunting Outputs

SANS Summits

<u>Measurements:</u>	<u>Outcomes:</u>
<ul style="list-style-type: none"><li>- # Hunts <b>executed</b></li><li>- # new detections <b>created</b></li><li>- # new detection <b>methods</b></li><li>- # detections <b>tuned</b></li><li>- # hosts <b>hunted</b> against</li><li>- # hosts <b>investigate</b></li><li>- <b>Dwell</b> time</li></ul>	<ul style="list-style-type: none"><li>- Documented internal anomalies\knowledge</li><li>- Identifying missing\required telemetry</li><li>- New TI acquired<ul style="list-style-type: none"><li>- Atomic Indicators</li><li>- Behaviors \ TTPS</li></ul></li></ul>

As an Analyst I like to see the OUTCOMES, and if you're capturing anything this is probably what it looks like today, data and information relevant to you and your team. There's not much tangible here for a pointy haired manager, someone whom you may need to justify your EXPENSES too. That's where the measurements come into place. The measurements are great as they highlight the failures as beings successes in this case. When Threat Hunting you often hope not to find anything and as such your activities and precious time may come up with big fat ZEROs for reported incidents. After spending lengthy time fighting with your tools, refining your queries, and gathering datasets wouldn't you still like to have some WINS besides not finding badness.

Capturing the metrics are easy to do and a light lift overall.

Now next time someone in your management change asks about the latest attack to hit the news and they ask can we detect that... Maybe you'll have already executed a hunt looking for one of the particular TTPs and you can tell them YES! We executed a hunt on XYZ date which covered (99% of our assets OR 67,000 hosts) we investigated 111 individual machines and implemented a custom detection for that TTP on XYZ date. Even if you don't have detections for the entire ATTACK chain you can still articulate what you have and where you'd detect the adversary in their campaign etc.

Directly relating your work and efforts in quantifiable numbers, with documents outcomes, and evidence you can use when relevant will allow you and your team to clear demonstrate value and continue to build, run, or expand your Hunting operations.

If you find nothing else useful from this talk take this one to HEART. You and your team are out there doing AWESOME work make sure you get credit for it continuously :) Just because you turn up empty handed doesn't mean you aren't bringing a tremendous amount of value, you just have to communicate it in a language your management can understand NOW get out there and prep some of that management eye candy (metrics, graphs, charts, tangible data) they'll love :)



I had a lot of trouble capturing this one with a one-liner! Honestly, here's the deal... I've yet to see decent reporting for Threat Hunting from a service provider, customer, or internal teams. You're asking for TIME or MONEY or BOTH in order to go off and perform an activity that you hope yields ZERO RESULTS. Are you going to turn in a blank report after hours, days, or weeks? I'd hope not but all too often the output I've seen has been extremely underwhelming if there even were any outputs outside of a short conversation or some basic documentation.

**YOU NEED TO BE SELLING THIS AND CAPTURING THE VALUE IT BRINGS TO YOUR ORGANIZATION!**

Threat Hunting is literally black magic buzzwords CISOs use to sell your efforts internally, but are you actually accurately capturing your body of work in a format that your management can grasp and understand? Can you break down the fruit of your labor into something management can digest? Is your service provider charging you for a service but not capturing and reporting how those "consulting hours" are being used and how these services are actually increasing your detections strengths? Probably not!