John Lambert

Defender’s Mindset

Nov 21·13 min read

Original: Defender’s Mindset. This is a collection of thoughts… | by John Lambert | Nov, 2021 | Medium

This is a collection of thoughts, quips, and quotes from tweets, blogs, and presentations over the years. If you find them helpful, drop me a note at @JohnLaTwC or on LI.

Attackers seek to turn illegitimate access into legitimate access

Your network often provides all the accesses and capabilities the attacker needs — because after all you need to manage the network too. If they obtain your legitimate credentials, they can use the tools and means you have put in place to achieve their goals. So while malware and exploits may play some part in their toolkit, attackers are just IT with different goals.

https://twitter.com/JohnLaTwC/status/682351553240403968

Prevention is the guardian of detection. Prevention creates the whitespace to detect and respond to the most important things.

A point with this one is that it's easy to get lost and overwhelmed in a noisy network. Your SOC will be flooded with alerts and won't

have the time to find the threats that truly matter. Your prevention approach (whitelisting, asset management, patching, identity management) can lower the noise level and give defenders the whitespace they need.

https://twitter.com/JohnLaTwC/status/682351869641949185

Biggest problem with network defense is that defenders think in lists. Attackers think in graphs. As long as this is true, attackers win

This is meant as a call to action to defenders to see their network as attackers do — as a set of nodes connected by control relationships and dependencies. Credentials give you access to nodes. Elevated credentials give you power over them. In turn this can unlock additional credentials due to stored credentials on the

node or give you control over other nodes due a security dependency. This creates a graph of connectivity.

Attackers land somewhere in the graph by spearphishing or finding an exposed server. Hacking is pivoting around the graph. Exploiting vulnerabilities or obtaining master secrets that allow one to forge identities is tantamount to creating new edges in the graph. While networks can be massive, careful study of a graph may reveal a small number of nodes or credentials that allow dominance over the graph because of the chain of pivots they allow. Tools such as Bloodhound help you do this kind of study.

This graph is often different than the mental model that IT uses to manage a network. Assets tend to be managed in classes: end user systems, servers, development and test systems, network appliances, and more. A lattice of access is created when devices are managed by different organizational departments across server management, vulnerability management, backup and BCDR, helpdesk, compliance, engineering, and security. Beware the web you create and know the graph.

Shared/Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win.md at master · JohnLaTwC/Shared (github.com)

https://twitter.com/JohnLaTwC/status/549049397602828288

Your network is a directed graph of credentials. Hacking is graph traversal. See the graph or all you'll see is exfil.

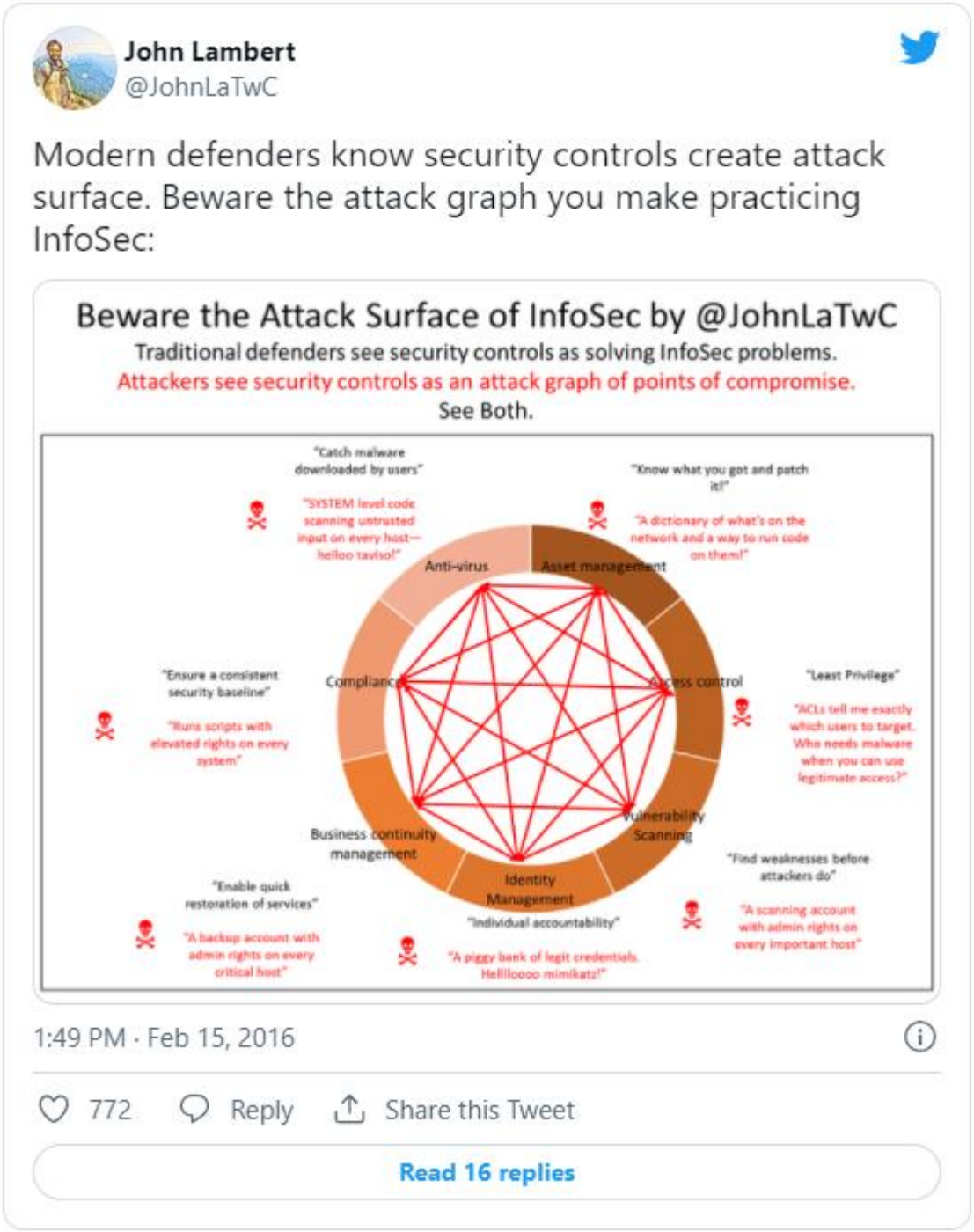
Each host on your network contains a set of credentials that can be stolen from it — credentials in memory, saved in files, stored in browser cookies, and the like. These credentials grant access to other systems. You can model this as a directed graph. Lateral movement is navigating the graph. Dumping credentials gives you more paths to follow. Every time you log onto a host in a way that leaves a credential in memory, you just changed the connectivity of the graph, making it more connected than before. Understand the dynamic nature of this graph, and you'll see a network the way an attacker does.

https://twitter.com/JohnLaTwC/status/549048749347975170

Modern defenders know security controls create attack surface. Beware the attack graph you make practicing InfoSec

This statement and accompanying infographic and blog are about how IT controls are not made of different stuff than the infrastructure they are protecting. IT management tools and DevOps pipelines are composed of infrastructure themselves. They have accesses, dependencies, credentials, and attack surface themselves.

https://twitter.com/JohnLaTwC/status/699304590500634625



A choice of technology is a choice of attack surface

Technology is not a standalone set of bits or interfaces. It is code that comes from somewhere — a build system, a release pipeline, and a team of developers. Developers make choices about whether the defaults are secure and make changes as the technology goes through its lifecycle. It is hosted on some infrastructure for its consumers. That infrastructure is maintained by someone, often a different organization than the one who built the technology. The dependencies stack up quickly. A point is that there are many controls needed to keep technology secure and it represents an attack surface that must be defended. Each technology comes with its own spiderweb of attack surface. Don't fear it. Understand it and make informed choices.

I gave a talk at VirusBulletin in 2018 where I spoke on this:

Shared/2018-10-VirusBulletin_JohnLa.Final.pptx at master · JohnLaTwC/Shared (github.com)

What is the most important network security spend: Sensor appliances? SIEM? Threat intelligence feeds? It's your analyst team.

It is common to see infosec budgets focus on appliances, threat intel feeds, and tools. There is nothing wrong with that but ultimately those tools are used by analysts. They are the ones investigating alerts. An investigation can hit a point of diminishing

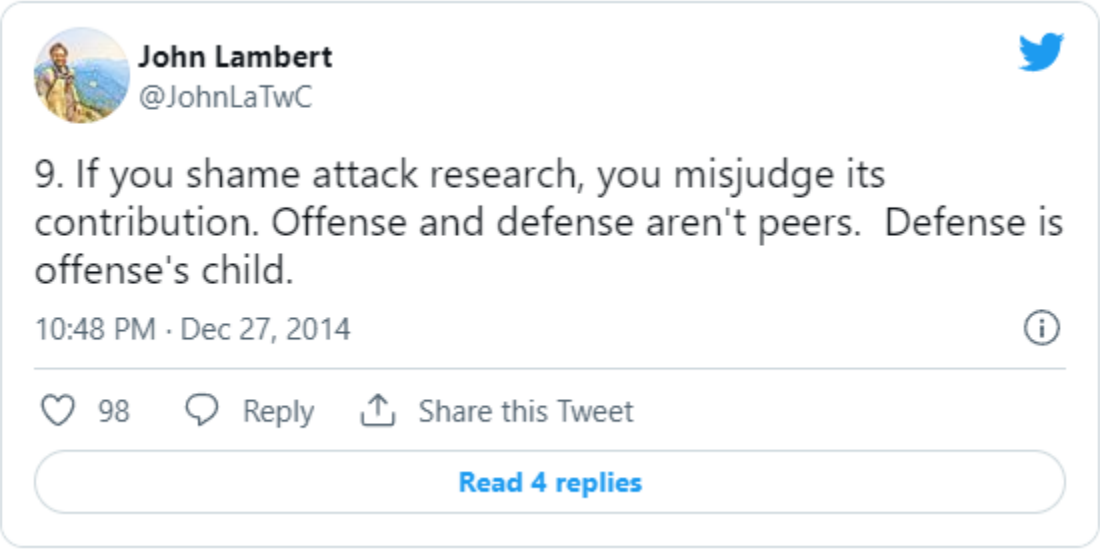
returns, but should you instead keep going? These crucial decisions fall to your analysts. Investing in them is not just about salary. I have yet to meet someone in infosec that doesn't want to improve their skills. Find ways to have them make connections with peers in industry, learn new skill domains they are excited about (reverse engineering, redteam, forensics, machine learning, big data, Splunk). Investing in them means ensuring they have predictable downtime to spend outside of work on their personal pursuits. Create whitespace to allow for projects that need additional time to get traction. Improve the baseline sensor system and environment so they have much better data to work with.

https://twitter.com/JohnLaTwC/status/549048664543350784

If you shame attack research, you misjudge its contribution. Offense and defense aren't peers. Defense is offense's child.

Some thought this tweet was a form of offense worship. Others pointed out that offense and defense co-evolve. My point is that defense starts from a mindset of protecting something (a system or a quality) from abuse or harm. I think it's very important for defenders to have an attacker's mindset too, otherwise we can fall victim to thinking that is ignorant of countermove. Defending means thinking about side effects from your defenses — defensive controls have attack surface too. It also means anticipating what the attacker will do next, whether you have a solution for that, and what the work factors are on the offense and defense side to see if you've truly moved the needle.

https://twitter.com/JohnLaTwC/status/549049324571611137



On vulnerabilities: You can argue over exposure, difficulty, and likelihood.

Security researchers write exploits because they like the truth.

Developers can spend a lot of time analyzing bugs to understand if they are truly exploitable. Can an attacker reach this code, is the race condition winnable in practice, is the way something crashes just a nuisance or exploitable? I remember one bug a pentester found in decompression code written in assembly language where the code missed popping a register in an error branch. The dev owner blew off the bug because he didn't think it was exploitable. That pentester spent the entire weekend to build a working exploit. When analysis is in doubt, exploits are truth.

https://twitter.com/JohnLaTwC/status/549048929753382912

Defenders, you're not stopping attacks. You're increasing attacker requirements.

‘Stopping’ breeds a mindset ignorant of countermoves.

Here the point is to think about how attackers will respond to your actions. You put up a firewall to block inbound attacks, and they go after endpoints through spearphishing. You deploy whitelisting, and they live off the land with LOLBAS. You deploy multi-factor authentication, and they steal tokens from endpoints that have already authenticated. Prevention is valuable — it raises the bar on many kinds of attacks and quiets the network so defenders have whitespace. But be ready for countermove.

https://twitter.com/JohnLaTwC/status/549048694956257280

Pentest is the most misused security practice. Pentest is diagnostic. Go from

treating the bugs as output, to treating them as input.

This is lamenting the fact that all too often a pentest is used as either a report card or compliance check-box. Pentest is diagnostic. What's in scope matters. What techniques are tried and applied matters. How findings are characterized matters. Use the results from pentest in a continual improvement program. It's not output — it's input.

https://twitter.com/JohnLaTwC/status/549048975328677888

Software engineers hide reality by using architecture over implementation. Hackers reveal reality by using implementation against architecture.

Software development uses abstraction to simplify and reason about itself. Bugs show up in the cracks of implementation, often in the boundaries between components. Security researchers know it's all a complex state machine which can operate in unintended states with the right input.

https://twitter.com/JohnLaTwC/status/445266529475506176

What fuels investment in security beyond compliance? Findings from IR, industry peer incidents, and proactive mgmt. Usually in that order.

Often the way you know whether your defenses are adequate or not is revealed by incident response. IR findings are truth. The opportunity is that you use that truth to motivate improvement. The peril here is that if your detection story is not adequate, you'll

not know you're being victimized and miss out on those IR truths. Incidents happening to others can be motivating as well. Smart people learn from their own mistakes, but wise people learn from the mistakes of others so they don't make them in the first place. Lastly, good management will self-assess and make improvements. The reason for the order I gave is that you can't invest without limit, so there is usually some real contact with reality that motivates it.

https://twitter.com/JohnLaTwC/status/549696321079111680

Attackers use your infrastructure. Make it a sensor with event collection. It's not the bite that makes a spider successful–it's the web.

Adversaries enter your infrastructure to pull off their attacks. They obtain your credentials to access systems, perform lateral movement to pivot around nodes, and use your network to take things in and out of it. Select the techniques you're most concerned about, enable the appropriate event logging and collection at scale, and index it so analysts can perform powerful queries and write rich detections. Locard's exchange principle is "every contact leaves a trace". Your magnifying glass is a query prompt. Turn your network into a set of microphones wired to floodlamps.

https://twitter.com/JohnLaTwC/status/676033863315881985

Go from sensor silos to a defender's pyramid. Link them and discovery is pivoting around the bases, exploiting leads.

There was a time in infosec when triangles, pyramids, or other models were all the rage. So it is here! The idea behind this pyramid is that are data sources which are linked to each other and defenders often join and pivot across them during investigations. Attackers use credentials as they access endpoints. Those endpoints contain valuable logs where the action occurred. Ingress and egress traffic tell you about penetration, command and control, and exfiltration. Hits and correlation from security detections inform you about threats.

Investigation is a set of steps that dip into these data silos and pivots among them. A security alert on a host may implicate a compromised credential. Logons records for that account may lead to another host that was the source of the activity. An examination of activity at the time of the attack may reveal a backdoor. Egress traffic from that backdoor may point back to a command and control server. Threat data about that C2 may tell you which adversary you face.

Data silos need to be linked. Use the pyramid model to link your most common data sources so selectors from one become join points with another. Happy hunting!

https://twitter.com/JohnLaTwC/status/572185917902991360

Adversaries need credentials more than malware. Deny them by avoiding the sins of Windows credential administration.

Sins of mirror imaging: You build a network so it's manageable. This manageability is often all the adversary needs to accomplish their goal. The mental model IT has on how they manage systems may differ a lot from how the systems can be managed by the credentials in reality.

Sins of abdication: Every system has local accounts. Sometimes many systems will have a local admin account (maybe helpdesk) with a common password. If an attacker learns that on one host, they can use it on many others. Local account logons leave no footprint on the domain controller because they are machine to machine. Hence, central visibility is not possible without agents on every endpoint.

Sins of tradeoffs: Expiring credentials can disrupt services or users. If you lengthen the expiry period, attackers get the benefit of this policy as well when they steal those credentials. Local admin passwords never expire by default. A credential stolen years ago will still be valid the next time they are in your network.

Sins of incompleteness: Think of your network in terms of equivalence classes. To protect an important server, you must equally protect the workstations where it is accessed by administrators. Failure to do so will give attackers a less defended way in.

Sins of wishful thinking: Most multi-factor auth systems have a notion of a logon and a session secret. If you're able to steal post authentication secrets and re-use them, the defender may have used MFA, but the attacker doesn't have to.

Sins of hygiene: Credentials are often stored in configuration

files, code, or documentation. Failure to store credentials securely provide an attack point.

https://twitter.com/JohnLaTwC/status/587289888560558080

Why poor infosec teams stay poor — understand the dangers of the Capability Chasm.

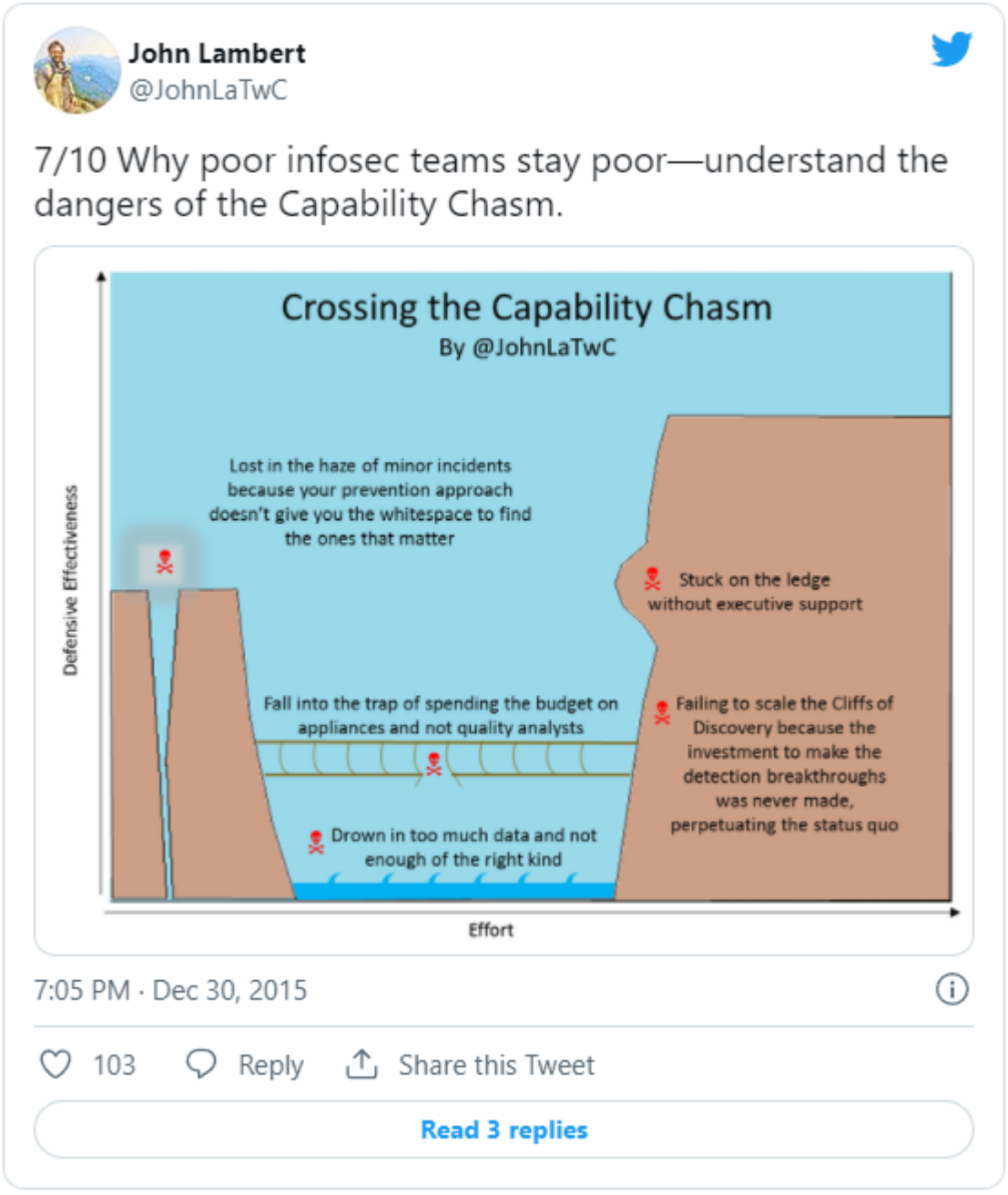
Defenders can be hindered from finding the attacks that matter by a number of causes:

- Their network is too noisy or poorly managed and their attention is sapped by too many incidents
- The budget was spent on solutions that don't provide the right bang for the buck in terms of discovery
- They have data but it's not the right kind. Their firewall logs ate up all their SIEM capacity so they have no endpoint visibility.
- A poor defensive posture makes it hard to attract talent which is capable of finding important attacks

A risk in perpetuating the status quo arises when teams fail to find "the big one" that proves to management serious delinquencies exist, and as a result they believe the current investment to be adequate. Often getting to the next level requires investment beyond the incremental, which are difficult for management to lobby for absent compelling evidence. Making room in an existing budget may involve unpopular choices or taking on sacred cows.

Getting that breakthrough, before a major attack has happened, is challenging and why this chasm exists.

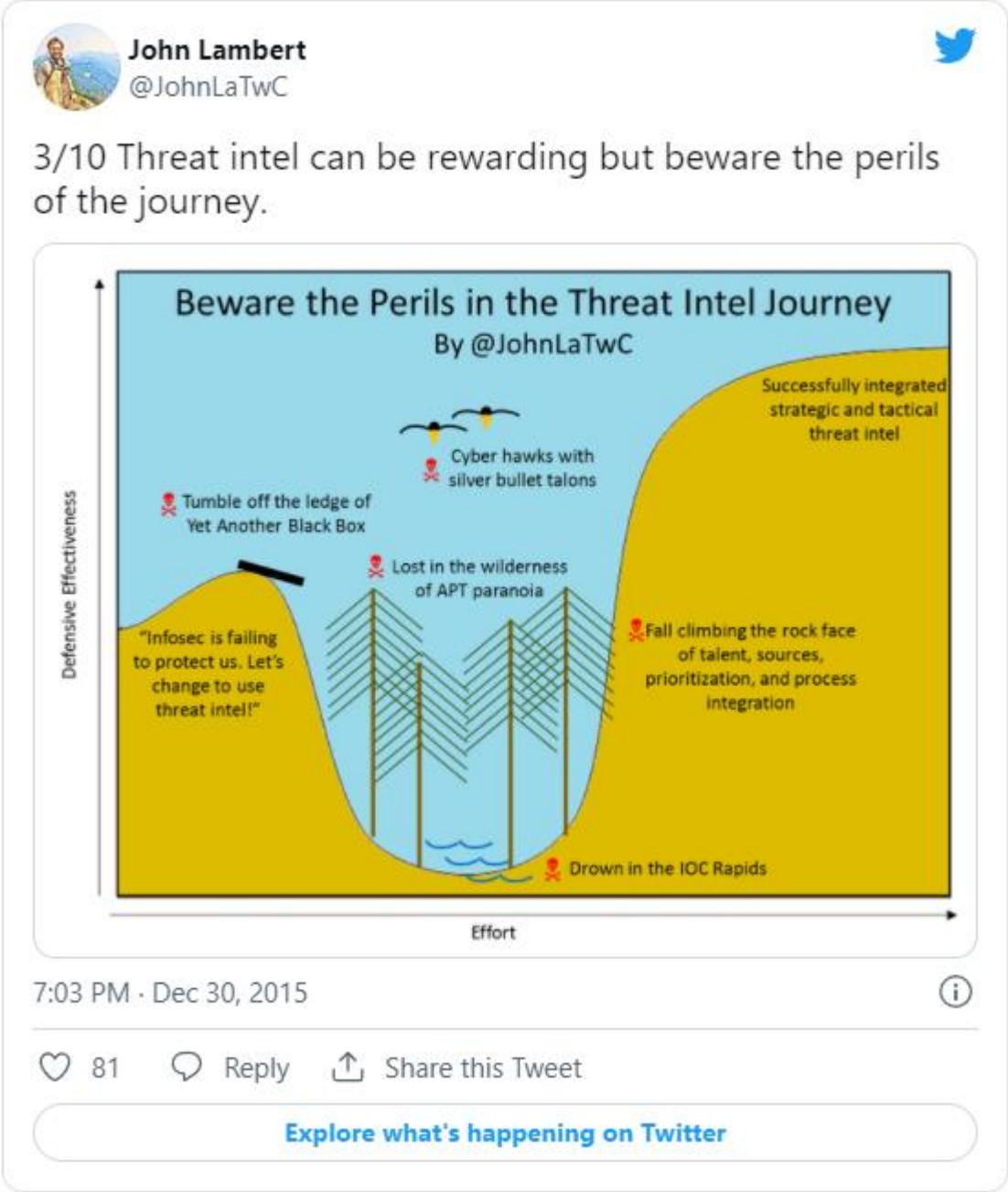
https://twitter.com/JohnLaTwC/status/682351790939967488



Threat intel can be rewarding but beware the perils of the journey.

Using threat intelligence can be a major breakthrough in defense. You can go from applying equal effort everywhere to spending time on the most critical threats. But getting there can be full of wrong turns. Done right, threat intelligence helps you prioritize, gives you context, and improves your decisions as defenders.

https://twitter.com/JohnLaTwC/status/682351253838413825



You realize that a successful anti-malware quarantine of a password dumper means your work isn't over — it just started.

While a password dumper is malware, how did it get there? Something or someone had the rights to drop and run the tool. Understand how that happened and you might find yourself in an IR.

https://twitter.com/JohnLaTwC/status/553587759374286849

The advantage of the attacker lies in the difference between what you have and what you manage

This was inspired by Rob Joyce's remark that attackers put in time to know your network better than you do.

https://twitter.com/JohnLaTwC/status/1437468407356477444

Quotes by others

Here are some other quotes that stuck with me over the years.

Do the third move first

I heard this from a person who worked at the NSA where it seemed to be institutional wisdom. This is about anticipating countermoves. If you do X and the adversary responds by doing Y, and you need to address that by doing Z, then do Z first. Anticipate.

An attacker is doing 2 + 2 to get 4 and that's a problem so you block it. Then they do 5–1.

I heard this from Neill Clift @clift_m when he was at Microsoft. This is all about knowing the degrees of freedom the adversary has before you mitigate. Examples often show up in areas where you're trying to reduce privilege. For example, removing the SeDebugPrivilege from administrators to prevent them from running tools that can read memory from sensitive processes. Administrators have wide ranging access to the system and have any number of ways to inject code into processes. Another example might be disallowing unsigned code on a system to prevent malware and attackers adapt by using LOLBAS.

You can get as close as yesterday

I love this quote by @ItsReallyNick is on adversary emulation. Threat groups evolve and change tactics. By the time a group's TTPs are known, they may be on to new techniques or using still undiscovered techniques. A redteam can emulate them. A blueteam can ready defenses against them. Just know you're fighting history's version of them.

https://twitter.com/ItsReallyNick/status/943844472600788992

General philosophy quotes

These are not infosec related, but as general concepts, they apply well to infosec.

If you want to go fast, go alone. If you want to go far, go together.

This is African proverb I used in my [2018 Githubification talk] at MITRE's ATT&CKCon. It really resonates with me as a defender and in just many other aspects of corporate life.

Beyond mountains, there are mountains

This is a Haitian proverb that is about the fact that when you solve one problem, there is another beyond it. You have to have a long term view and recognize that anything hard will require iterations to make progress. You can apply this to your career. In the beginning, mastering subject matter seems very important to get the excellence needed. As you become an expert you gain a greater understanding of the surrounding business and things seem less black and white. You also realize how important people and processes are to improvement. Build depth in understanding stakeholders, clear communication, achieving consensus, and bringing people along with you.

I am terrified of believing things that are not true

This one is from a blog by Halvar Flake.

** For every opinion, no matter how bizarre, it has become easy to find a community with similar beliefs.*
** The discoverability of almost all information coupled with the shortening of attention spans allows people with strange beliefs to search for information at — at least if only glanced at for 15 seconds — may be interpreted to confirm their strange belief.*
** Algorithms that maximize engagement also maximize enragement — if the algorithms show me content that draws me*

into a time-sink discussions with no benefit, they are "winning" (in terms of the metrics against which they are evaluated).
** The social media platforms favor "immediacy of feedback" vs. taking time to think things through*

My self-help guide to making sense of a confusing world

ADD / XOR / ROL: My self-help guide to making sense of a confusing world (addxorrol.blogspot.com)

In closing

I hope you found some of these quotes useful. If you found some of them oversimplified, that's somewhat the nature of quips. A few of them use the language of IT, but feel free to adapt them to the world of DevOps and serverless computing to see if the lesson still applies.

May you find enough truth to help you ponder a problem in a new light.

— John Lambert

Distinguished Engineer, Microsoft Threat Intelligence Center, @JohnLaTwC