This presentation presumes you have a Red Team, big or small. It does not explain how to start a Red Team.

--

--

--

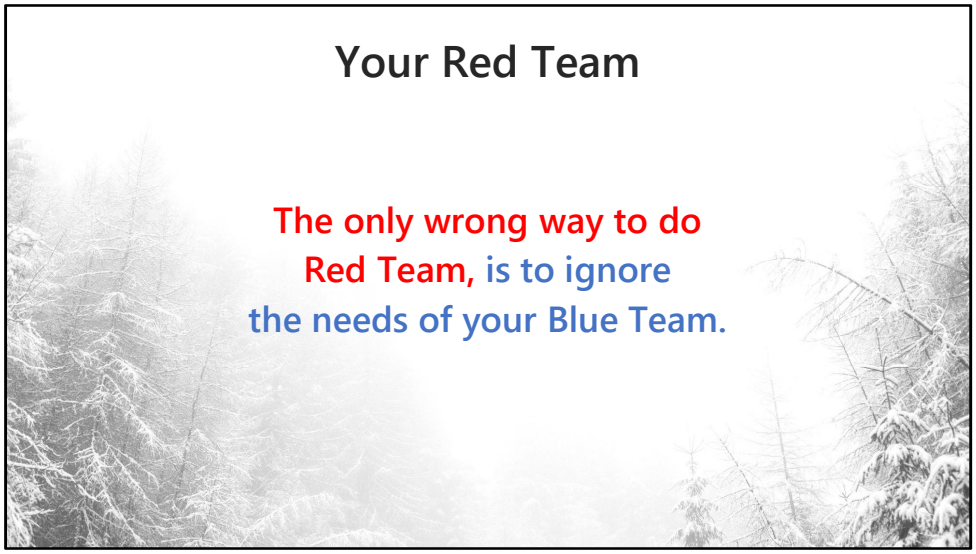
--

--

--

--

-end-



It must be said, though…

The only wrong way to do Red Team, is to ignore the needs of your Blue Team.

Our Red Team

- Seven engineers
- Varying backgrounds, experience
- Tightly integrated with Blue Team
- Share cool stuff, with cool people

We are seven engineers and one manager. We have grown and changed in size and shape over the years. We've been reorganized, and shuffled.

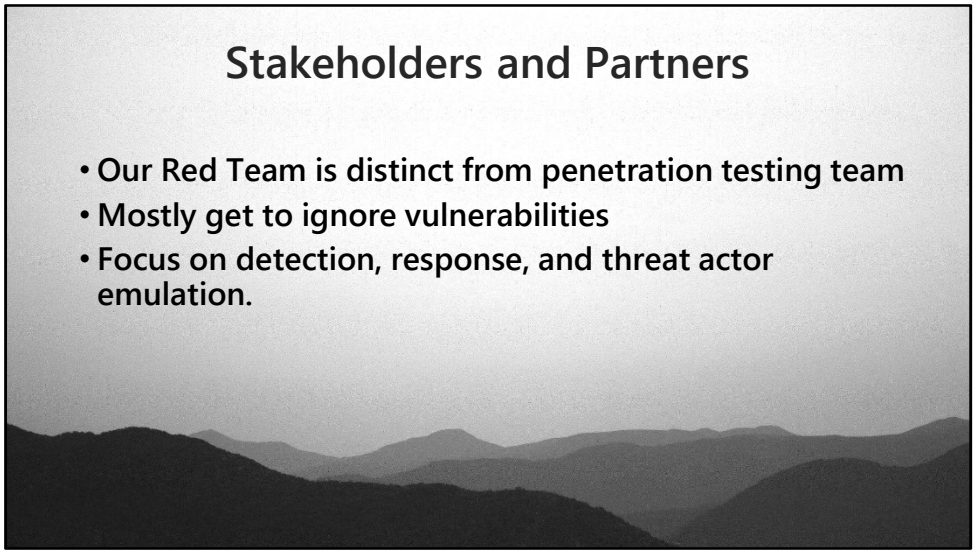
We have various backgrounds, and levels of experience.

We work alongside the Blue Team. We encourage both casual and formal collaborations. We even eat lunch together…

And every day we share cool stuff with cool people. People who are engaged, and challenge us, and make us want -- to make each other better.

We support and are supported by our incident responders, our detection engineers, our security engineers, our intel analysts, our various internal business teams, and our leadership.

We collaborate with product owners and business teams to understand how they get work done, and to improve their defenses against threat actors.



We are distinct in function from our penetration testing and vulnerability management teams, which, combined are several times larger than we are.

As well, their stellar performance prevents us from having to spend a lot of time hunting for and reporting on vulnerabilities.

This allows us to focus on detection, response, and threat actor emulation.

Our environment spans some 1,900 physical locations. This leaves ample opportunity for variation and drift.

Our security technologies, and some thousand manually maintained rules to support them, are spread across the enterprise.

Their functions are maintained by a variety of teams, including vendors, and IT teams like desktop engineering.

One team may install a technology, another may maintain it, and yet another may make sure it's running.

The equal distribution and effectiveness of all of these technologies has to be reasonably assured.

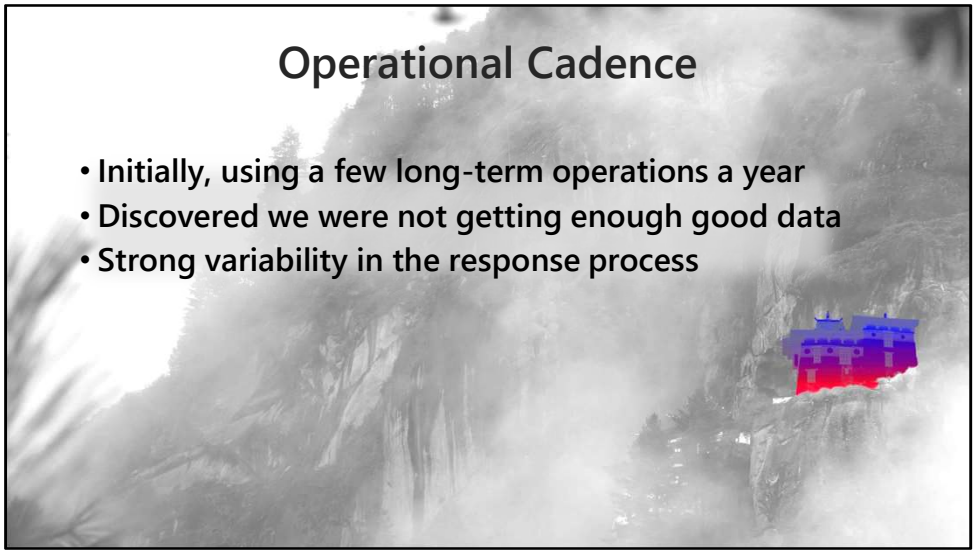
Our Red Team is a major driver of this goal.

In service of this, our Red Team gathers information from inside our organization in order to best prioritize the playbooks for attacking the environment.

The business deploys new technology at an astonishing pace, so we're constantly learning.

We combine that with the data collected by our extremely capable threat intelligence team.

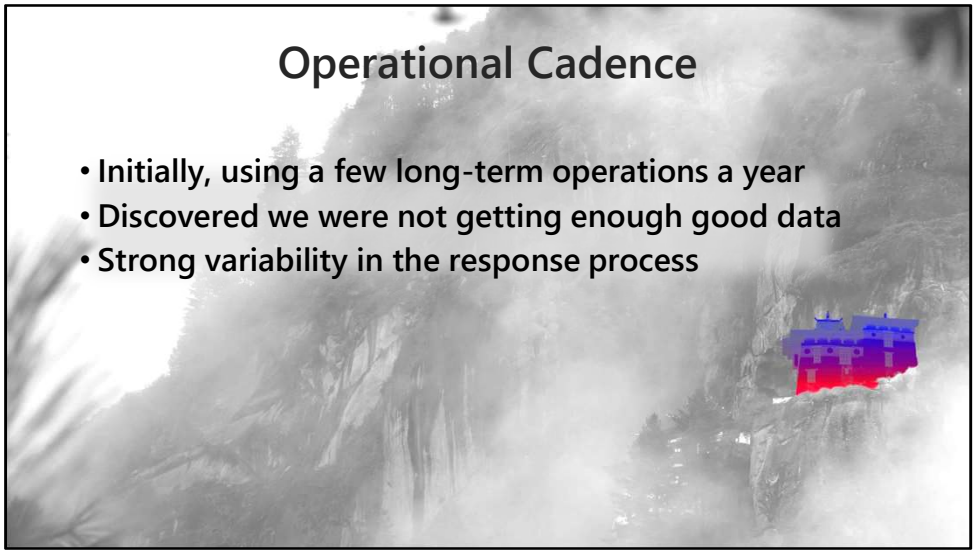
...



Like many other Red Teams, we were doing a handful of long-term operations a year.

However, there is a finite amount of data we can extract from each individual operation.

For example, the affected hosts and techniques. What was kind of surprising was that even the individual analysts assigned to the response had slightly different approaches to the work.



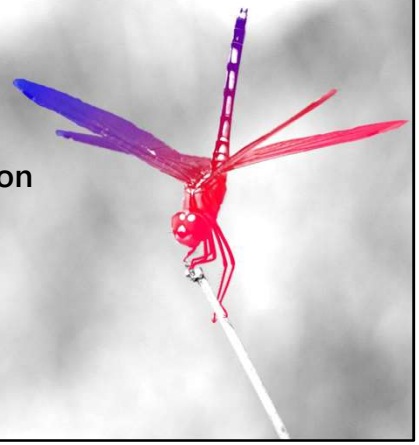
We found the greatest variability comes from the incident response process itself, rather than the environment.

You'd think that a very large, constantly changing network would be king of the unknown, but alas…

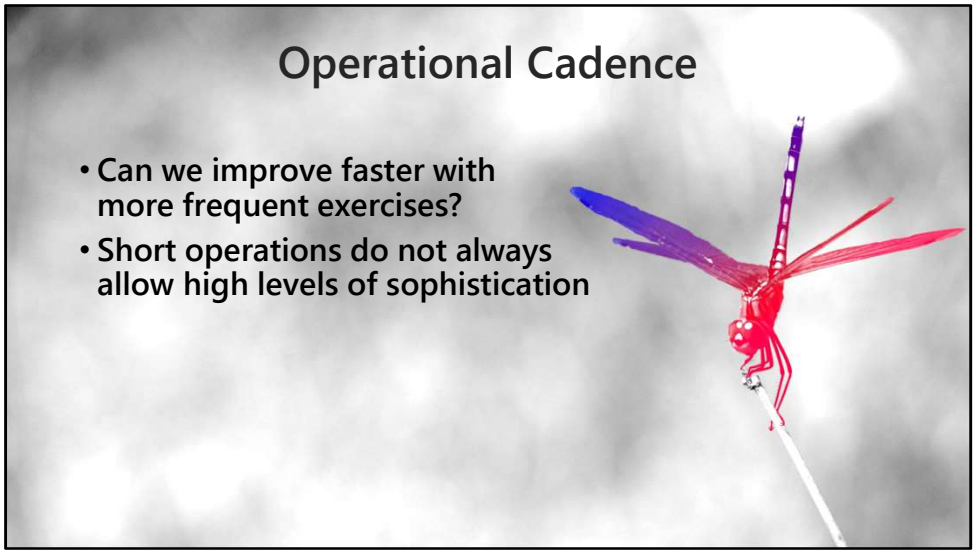
The response process provides us the most opportunity for improvement.

Operational Cadence

- Can we improve faster with more frequent exercises?
- Short operations do not always allow high levels of sophistication

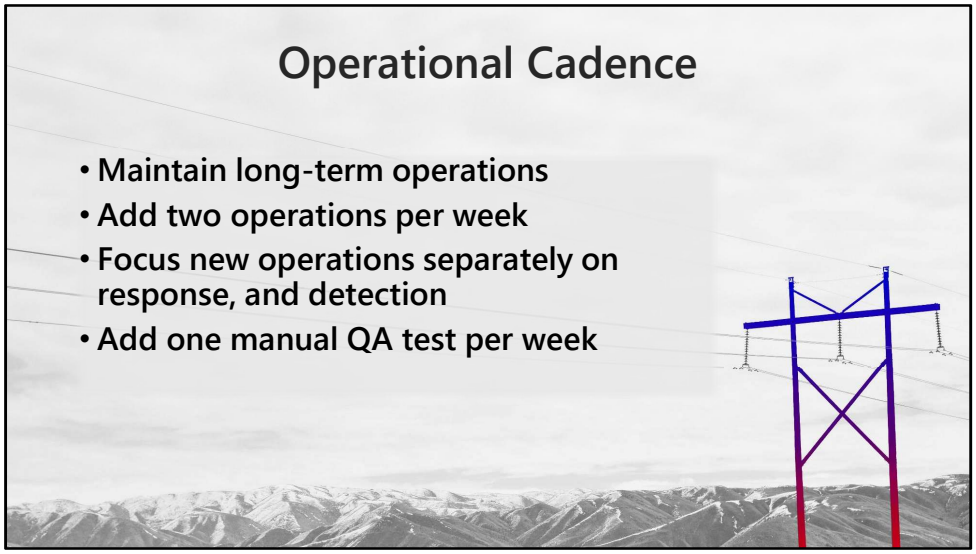


We came to the conclusion that if we want to maximize the amount of novel information our Red Team can create, we'd run shorter, more frequent operations.



We also acknowledged that short operations have a few disadvantages; among them that some threats may persist for months before discovery.

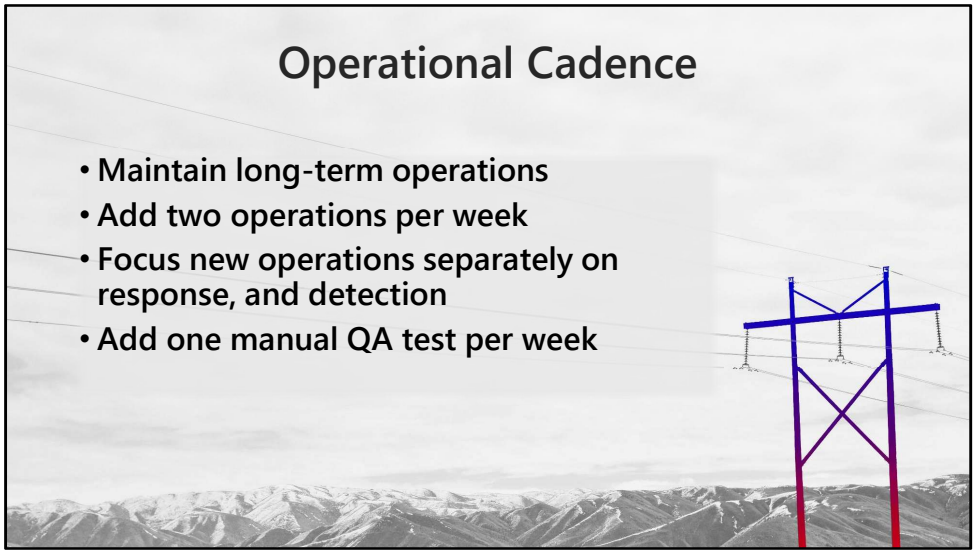
Adversary actions may become complex, and have a large blast radius. Accurately simulating those threats is difficult without adequate time.



In order to achieve an ambitious goal like one hundred operations per year, we first organized two weekly operations.

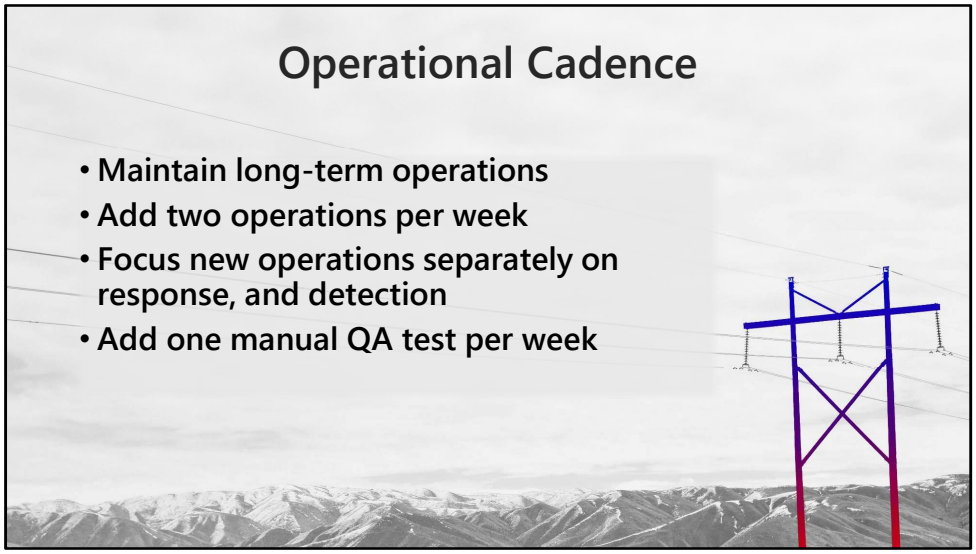
These had similar structure, such that every step of an attack was covered in both cases.

But the key difference was that one operation always completed with a technique that matched a detection rule.

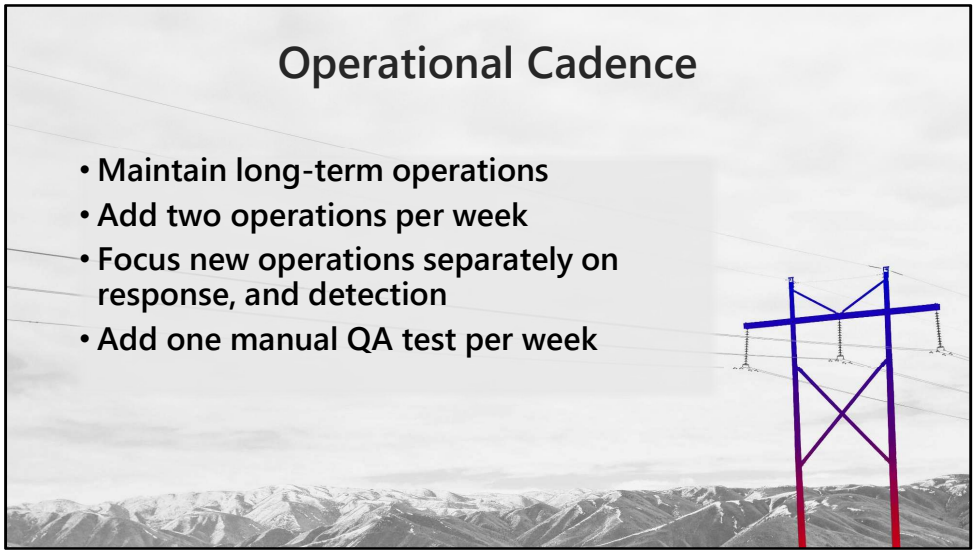


This operation would always intentionally fire an alert.

Yes, I hear you screaming colors at me.



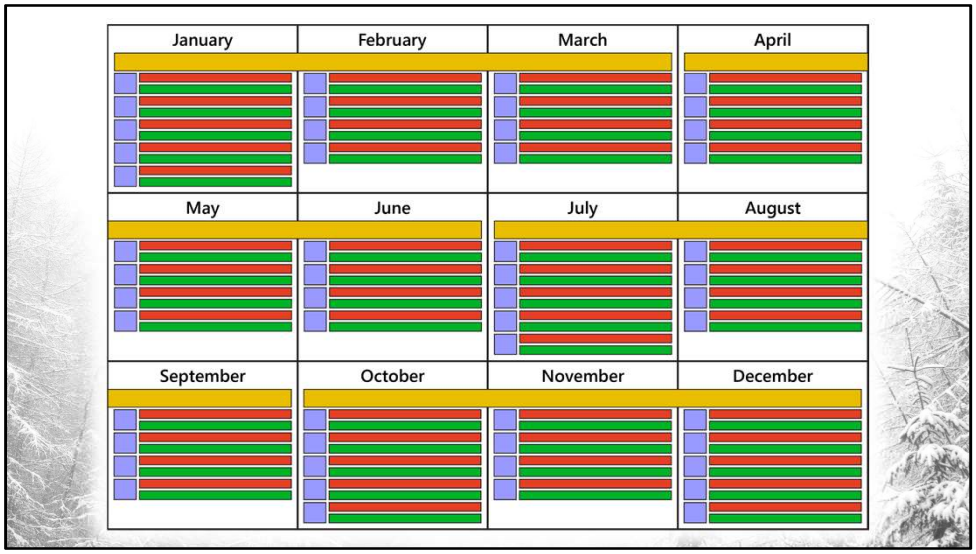
What this allowed us to do is scope certain operations almost entirely around response. We can then scrutinize the incident response process, rather than add compounding factors around detection and visibility.



The second weekly operation was more typical of other Red Teams, done mostly within visibility or detection gaps in order to achieve the objectives.

We also sought to answer questions like “Are all existing rules generating expected alerts from endpoints at all network locations?”

So, we added a weekly hands-on QA test for that purpose. That’s three projects a week.



All this meant we were on the hook for manually doing about 150 reportable exercises a year.

A number exhausting even to say out loud.

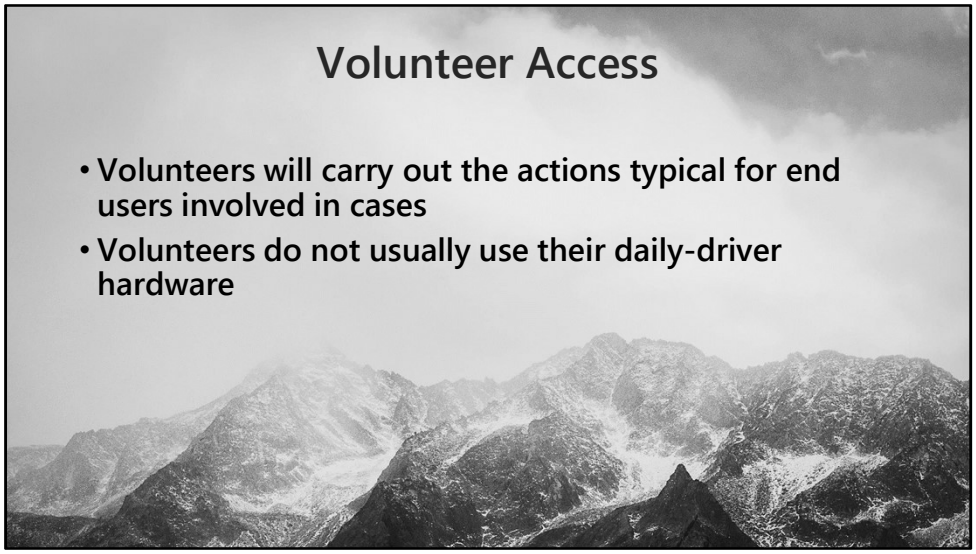
Now, how did we go about this?

Beginning each operation with an earnest attempt at initial access, we will have expended a lot of time and energy on a small selection of attacker techniques.

Consider this list of technique categories.

The value of our operations is primarily in techniques that occur AFTER initial access. This is where our most impactful improvements occur.

Given that understanding, our most frequent operations are executed with the assistance of volunteers from our business.



Those volunteers usually help us in executing a social engineering attack. We'll create a scenario, like an email. That volunteer will be asked to do what people are so good at doing, opening emails.

This work is also commonly done on check-out hardware, which is lent out to volunteers specifically for testing purposes.

Since our Red Team operations are frequently responded to, and the hosts and accounts scoped into those operations are sometimes disrupted by the incident response process.

We have to be cautious about how frequently we interact with production assets and productive people.

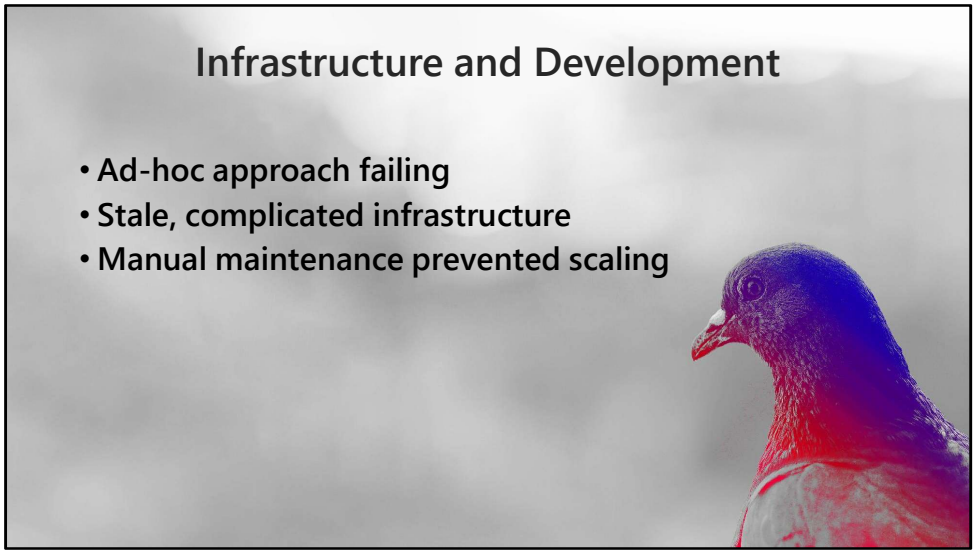
Only a modest percentage of the hundreds of operations we've done have had some business impact, and those impacts are almost entirely in end-user productivity.

For example, there may be a small disruption when an end user's password is reset.

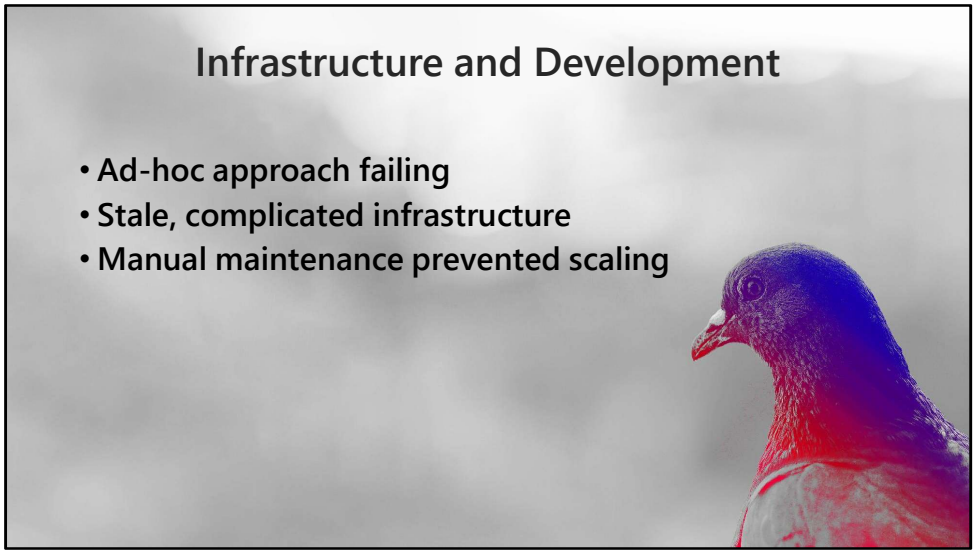
We've essentially traded some demonstrations of high risk techniques in sensitive production environments for far more frequent demonstrations of low and medium risk techniques in production-LIKE environments.

We get additional benefits by engaging more directly with team members in our business in a positive way, and bringing awareness of what our Red Team does outside of security.

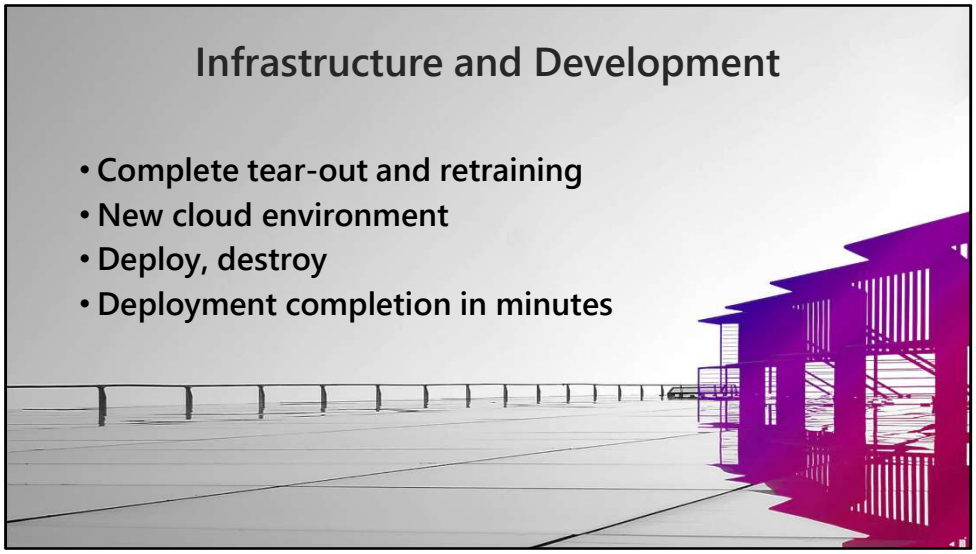
...



Taking an ad-hoc approach to development and infrastructure really held us back. It was taking several days to prepare the components necessary to run an operation, which meant a weekly test was stretching out well beyond the test week itself.



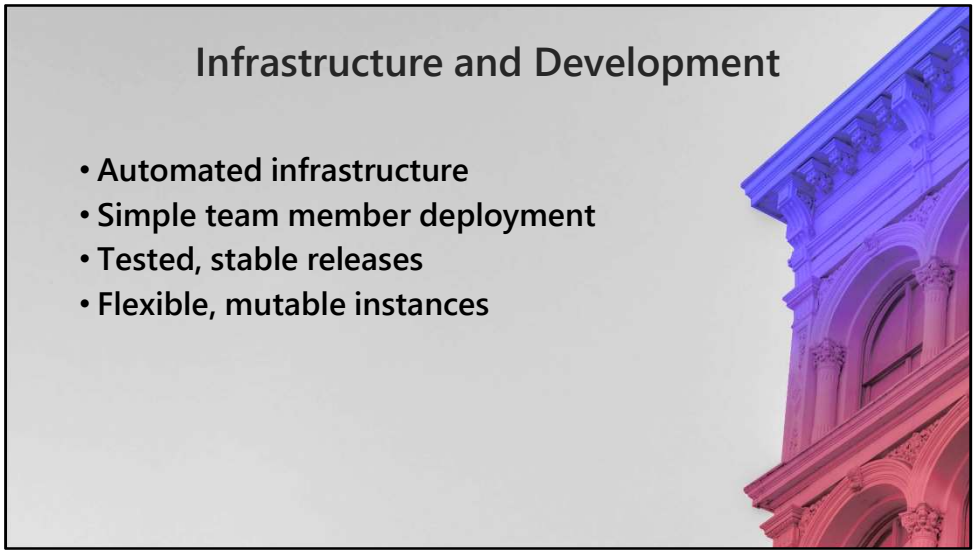
With multiple, concurrent operations, a setup where a few servers that live forever, but get constantly changed and repurposed would not let us scale.



We changed the way we approached infrastructure by doing a complete tear out.

We built from the ground up a new cloud environment where computing resources were automatically deployed as needed for specific operations, and then destroyed at their completion.

This deployment process takes minutes.



This was achieved using infrastructure and configuration automation, deployed from code repositories, and configured with a small text file.

While we have a single master branch for everything, all of our team members can deploy their own branches, or make changes either before or after deployment.

What we've managed to do with this is drive down the busy work, streamline tool development, and let operators focus on executing techniques, which is the core of our Red Team operations.

SO -- We do DevOps, managing our own code and knit it tightly with the underlying infrastructure.

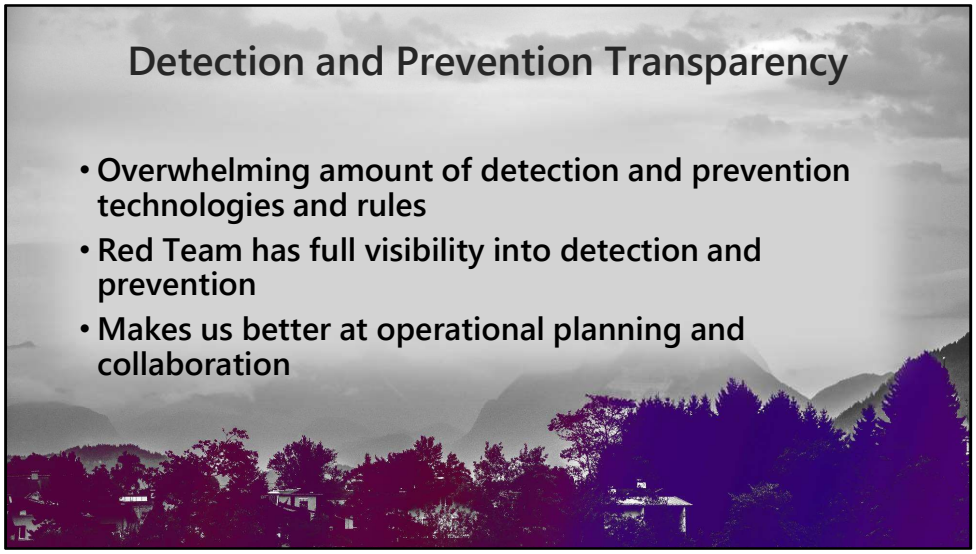
We plan our development out as best we can, track everything in issues, and use a project management tool.

But the security of our environment is the very first priority. Since security entirely builds around automated configuration, we can worry less about errors made by operators.

This may sound like a lot of overhead, but be sure we are very conscious of how much time we spend making tools.

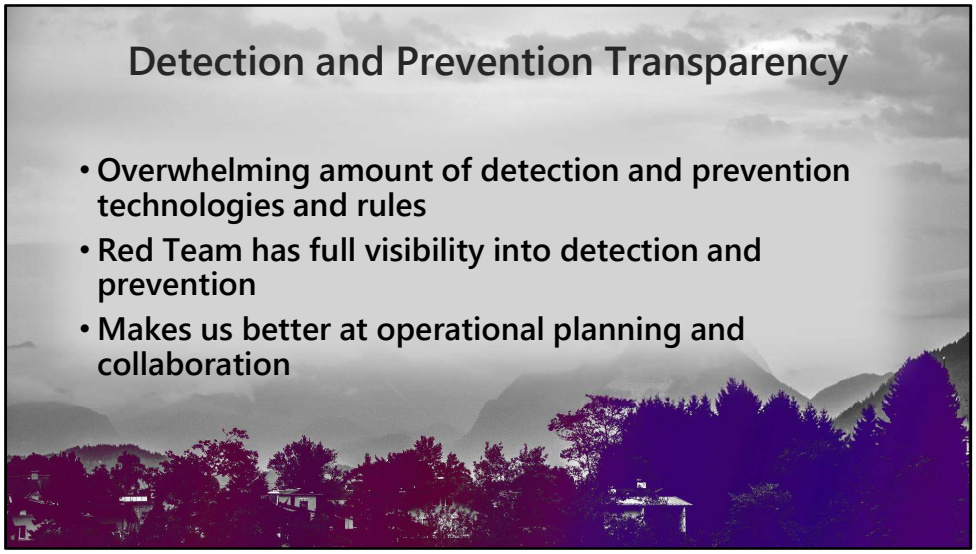
Now, If I write piece of code, I have to make sure someone who's never seen it before can run it safely during an operation, and, if they choose to, make changes to commit back to the repository. This means everything is documented.

With this process, we've managed to make tool making sustainable.



Maintaining detection awareness has been challenging. There are dozens of distinct technologies that our Red Team has to track where detection and prevention reside.

We as an organization don't even own some of these technologies. It may be hard or impossible to query a rule set that exists in a technology for that reason.



But whatever can be made available to our Red Team has been made available. We operate with complete technology transparency.

This SAVES US A LOT OF TIME. It also gives us opportunities to spot problems that thousands of operations might never reveal.

Our Blue Team has built out a tool which catalogs all of the rules in our environment, along with metadata about when they were created, where they live, and what intelligence they were based on.

We use this and several other tools to pick which techniques to use while planning operations. Our Red Team can reach into these tools, do a search on a keyword, and know most of what needs to be known about how well we as an organization can detect a particular technique.

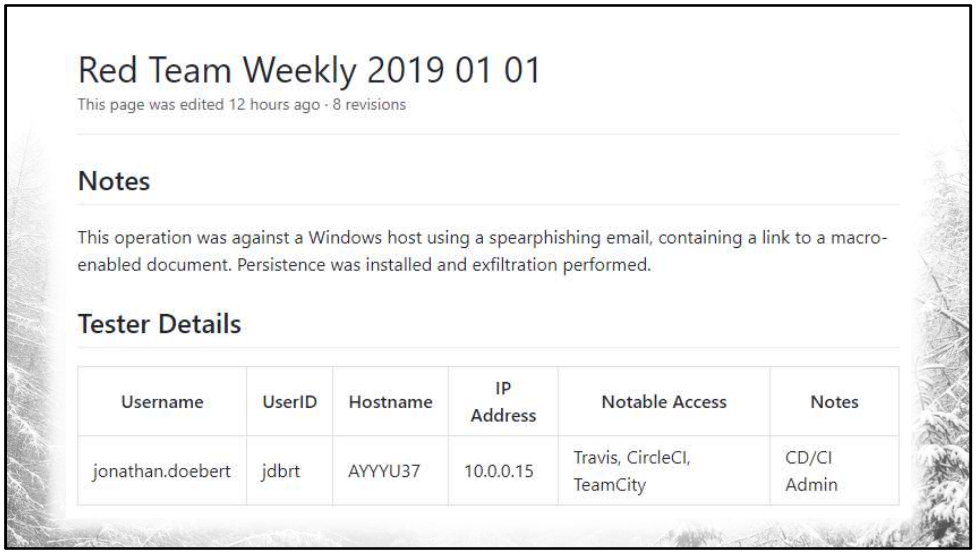
Our Red Team and Blue Team share the same physical space, and that allows for casual collaboration.

We also share a chat room specifically for Red-Blue collaboration, where information around rules and techniques can be shared.

Since we do it so frequently, we've tried to keep our reporting requirements to a minimum. Each weekly operation gets a very concise WIKI entry.

When the reports are completed, they are made available to a fairly large list of security team members, so everyone gets a sense of what Red Team is up to.

Long term operations may get a more verbose report.



Red Team Weekly 2019 01 01

This page was edited 12 hours ago · 8 revisions

Notes

This operation was against a Windows host using a spearphishing email, containing a link to a macro-enabled document. Persistence was installed and exfiltration performed.

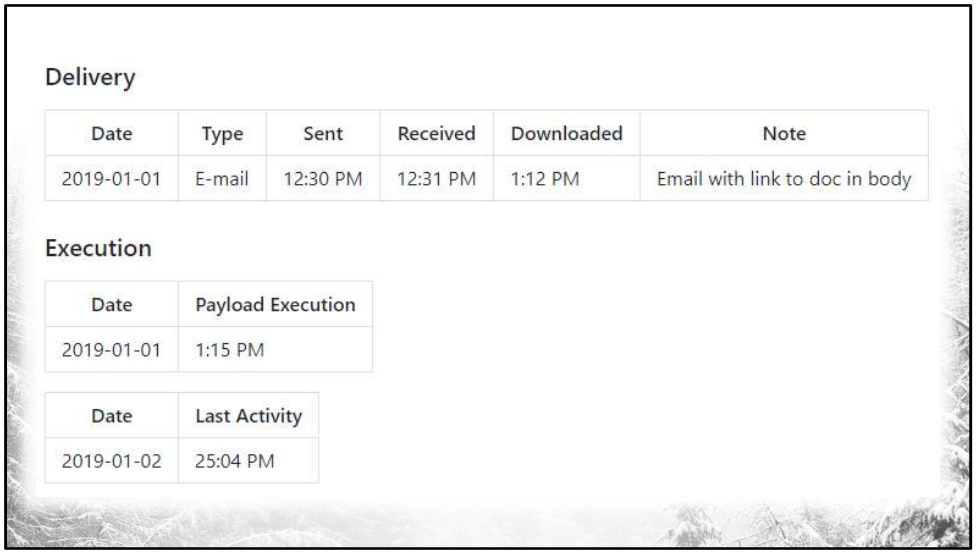
Tester Details

Username	UserID	Hostname	IP Address	Notable Access	Notes
jonathan.doebert	jdbrt	AYYYU37	10.0.0.15	Travis, CircleCI, TeamCity	CD/CI Admin

This is the template we use for weekly operations.

At the top, a short summary of the operation and its objectives.

We identify the subject or subjects of the testing. We also note what role and access they have in order to put the compromise into perspective in terms of risk to the organization.



Delivery

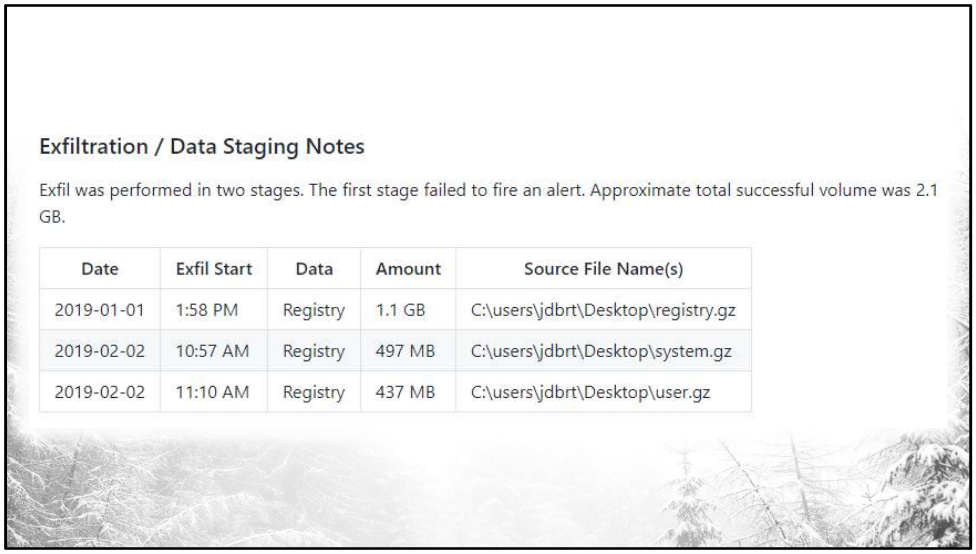
Date	Type	Sent	Received	Downloaded	Note
2019-01-01	E-mail	12:30 PM	12:31 PM	1:12 PM	Email with link to doc in body

Execution

Date	Payload Execution
2019-01-01	1:15 PM

Date	Last Activity
2019-01-02	25:04 PM

We'll document details about the delivery mechanism, and timestamp everything. This includes when compromise activity begins and ends.

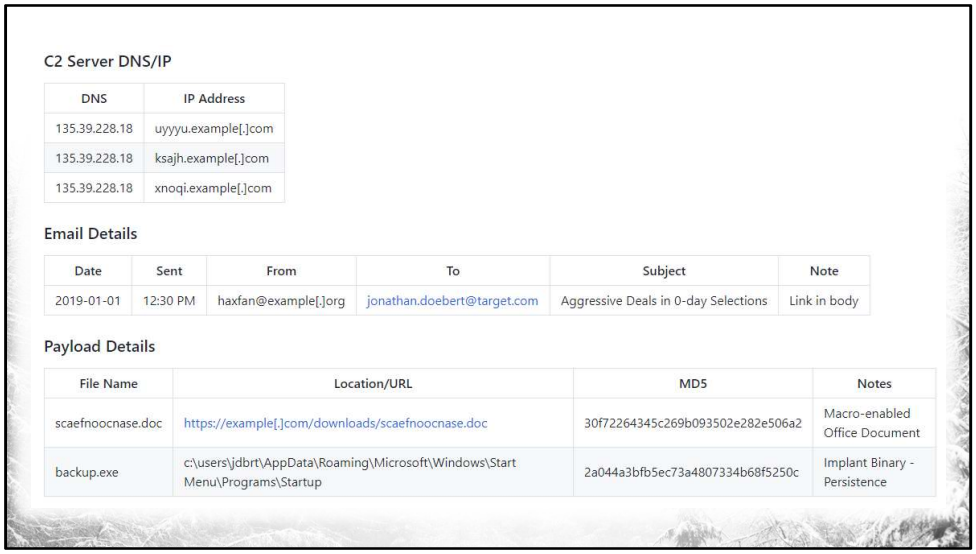


Exfiltration / Data Staging Notes

Exfil was performed in two stages. The first stage failed to fire an alert. Approximate total successful volume was 2.1 GB.

Date	Exfil Start	Data	Amount	Source File Name(s)
2019-01-01	1:58 PM	Registry	1.1 GB	C:\users\jdbrt\Desktop\registry.gz
2019-02-02	10:57 AM	Registry	497 MB	C:\users\jdbrt\Desktop\system.gz
2019-02-02	11:10 AM	Registry	437 MB	C:\users\jdbrt\Desktop\user.gz

If there’s exfiltration performed, we’ll identify how the data was staged, how much, and when it was done.



C2 Server DNS/IP

DNS	IP Address
135.39.228.18	uyyyu.example[.]com
135.39.228.18	ksajh.example[.]com
135.39.228.18	xnoqi.example[.]com

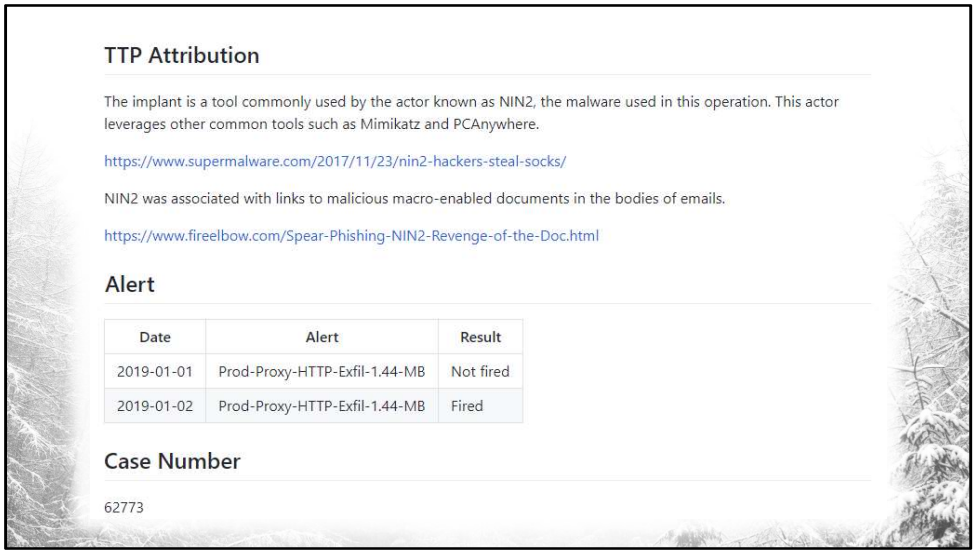
Email Details

Date	Sent	From	To	Subject	Note
2019-01-01	12:30 PM	haxfan@example[.]org	jonathan.doebert@target.com	Aggressive Deals in 0-day Selections	Link in body

Payload Details

File Name	Location/URL	MD5	Notes
scaefnoocnase.doc	https://example[.]com/downloads/scaefnoocnase.doc	30f72264345c269b093502e282e506a2	Macro-enabled Office Document
backup.exe	c:\users\jdbrt\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup	2a044a3bfb5ec73a4807334b68f5250c	Implant Binary - Persistence

We'll document specific IOCs, like hostnames, addresses, e-mails, hashes, and URLs.



TTP Attribution

The implant is a tool commonly used by the actor known as NIN2, the malware used in this operation. This actor leverages other common tools such as Mimikatz and PCAnywhere.

https://www.supermalware.com/2017/11/23/nin2-hackers-steal-socks/

NIN2 was associated with links to malicious macro-enabled documents in the bodies of emails.

https://www.fireelbow.com/Spear-Phishing-NIN2-Revenge-of-the-Doc.html

Alert

Date	Alert	Result
2019-01-01	Prod-Proxy-HTTP-Exfil-1.44-MB	Not fired
2019-01-02	Prod-Proxy-HTTP-Exfil-1.44-MB	Fired

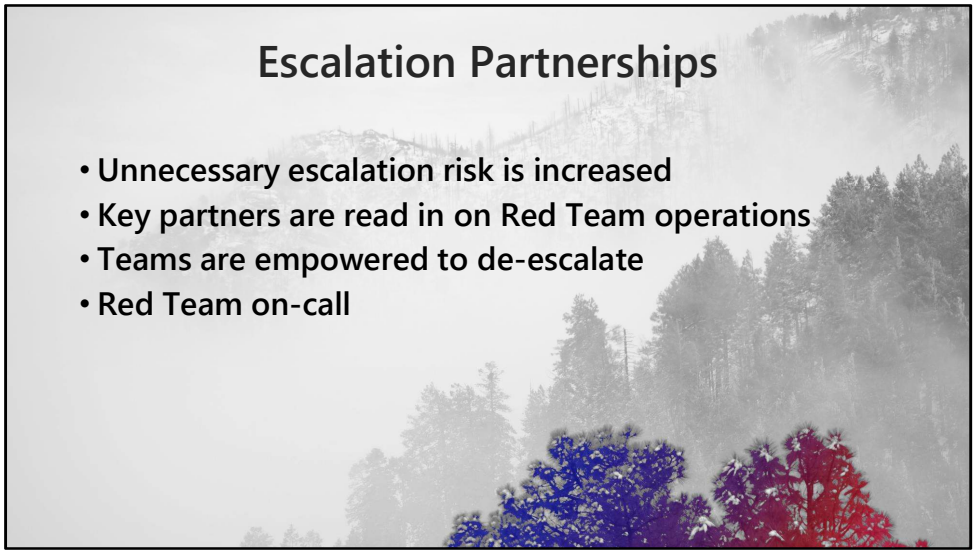
Case Number

62773

To tie our operational work to threat actors, we'll list out how each attributable technique is related to an in-the-wild threat actor, including a link to public or internal documentation on the subject.

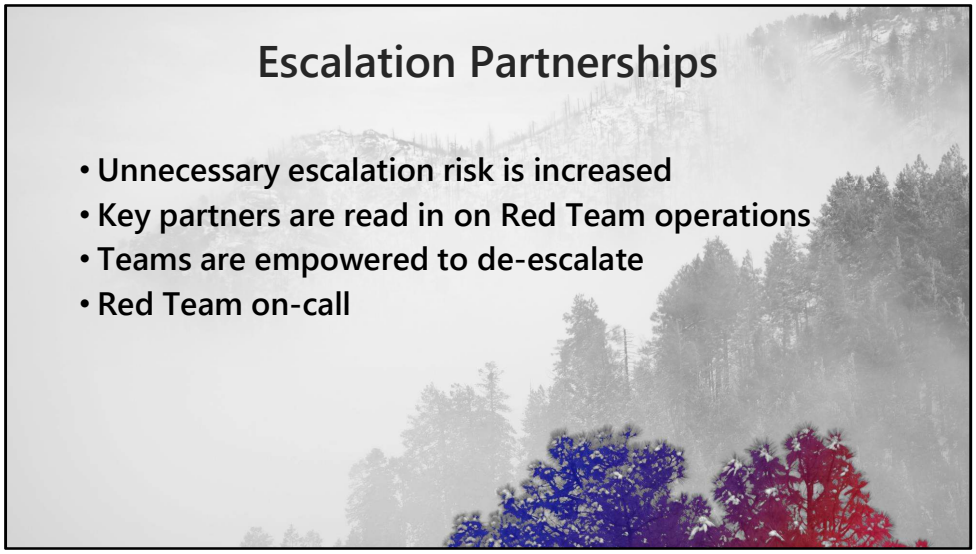
And we'll enumerate any alerts that should have fired.

This may seem light-on-details, but we do also keep the logs generated by our tools and servers.



With a high volume of operations, the opportunity for unnecessary escalations increases pretty dramatically.

When an operation begins, a selection of partners from the Blue Team are alerted to the timeframe and intended targets.



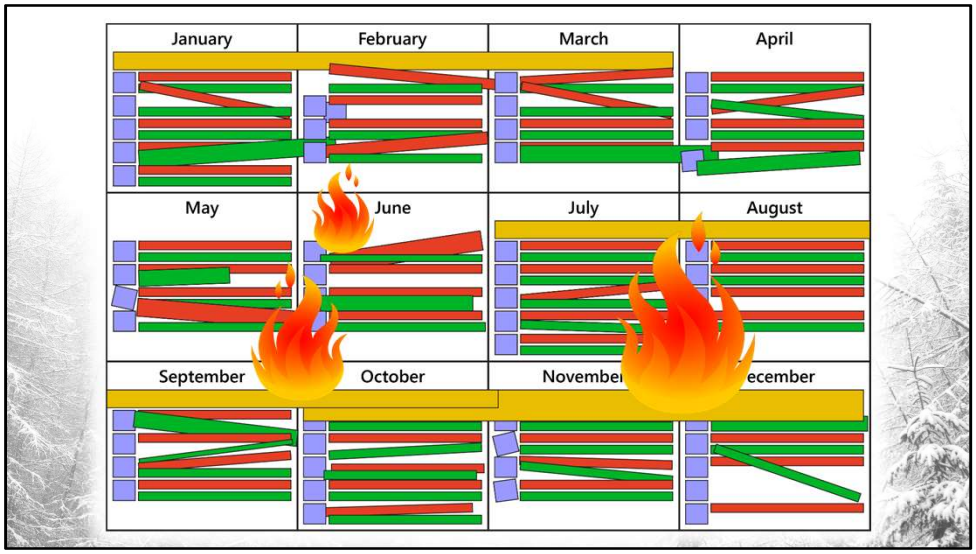
There’s an agreement to keep these data points secret from those conducting response activities.

That allows incidents to run their course while empowering the right people to de-escalate.

...

Eventually, the grind of executing basically three operations per week with only a handful of engineers got the best of everyone. Us on the Red Team, our detection engineers, our incident responders -- we were all a bit overwhelmed.

Our long-term operations were suffering from a lack of resources, and we weren't getting good value doing an additional weekly manual QA test.



Our ambitious plan to conquer improvement through brute force looked a bit like this.

Though, perhaps we might take what we've learned doing two labor-intensive operations a week, and incorporate them into a more balanced schedule.

To approach this, we now separate operations by time span and objective. We also take some of the work and automate that testing.

Operational Diversity

- A few strategically focused long term operations
- Two monthly detection and hunt focused operations
- A single weekly, response focused operation

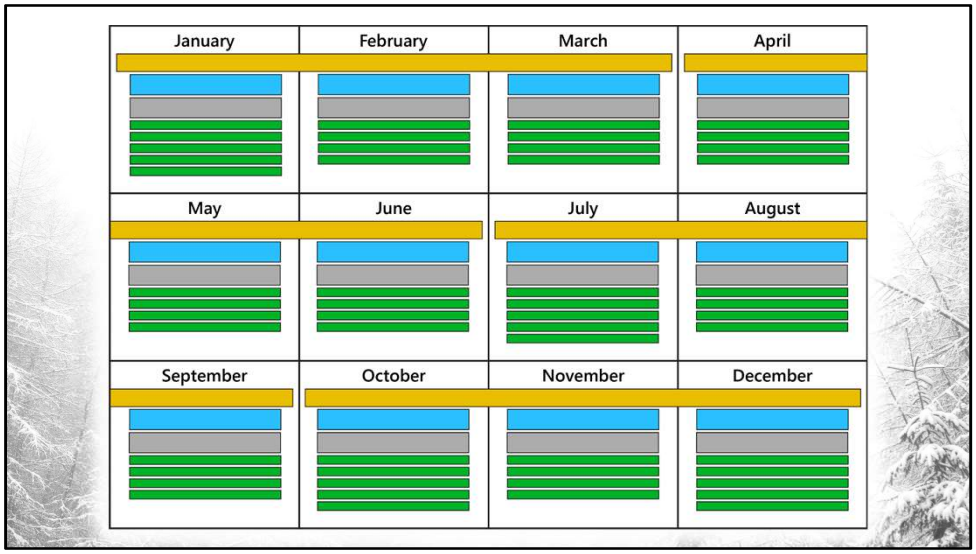
For us, long term operations are strategically focused, looking for gaps in process, policy, or design.

Medium term operations are centered around detection engineering, hunt capability, and individual business teams. We now run two of these a month.

Operational Diversity

- A few strategically focused long term operations
- Two monthly detection and hunt focused operations
- A single weekly, response focused operation

Short term operations primarily exercise response capability, and do some quality control on detection engineering. We cut those down to once a week.



Now we have wide coverage of different organizational goals, while maintaining high quality and quantity of results.

With automated or continuous QA testing, we can monitor the effectiveness of the existing detection and prevention technology.

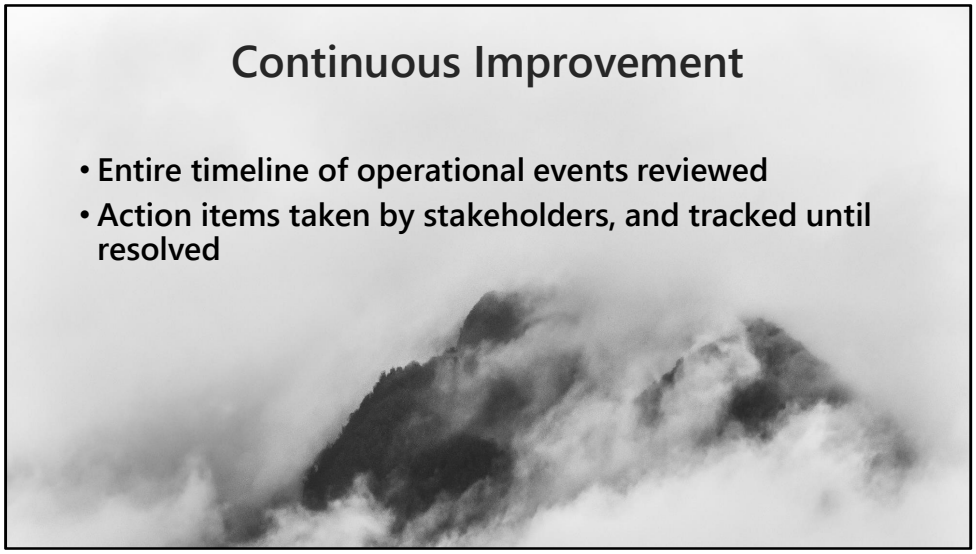
Our Red Team partners with other security engineers to develop this. We're hopeful that we'll have hundreds of tests running daily that require little-to-no human interaction.

Not only can we provide high quality intelligence-backed Red Team operations, but our Red Team is not burdened with continuously monitoring for drift or failure in the existing controls.

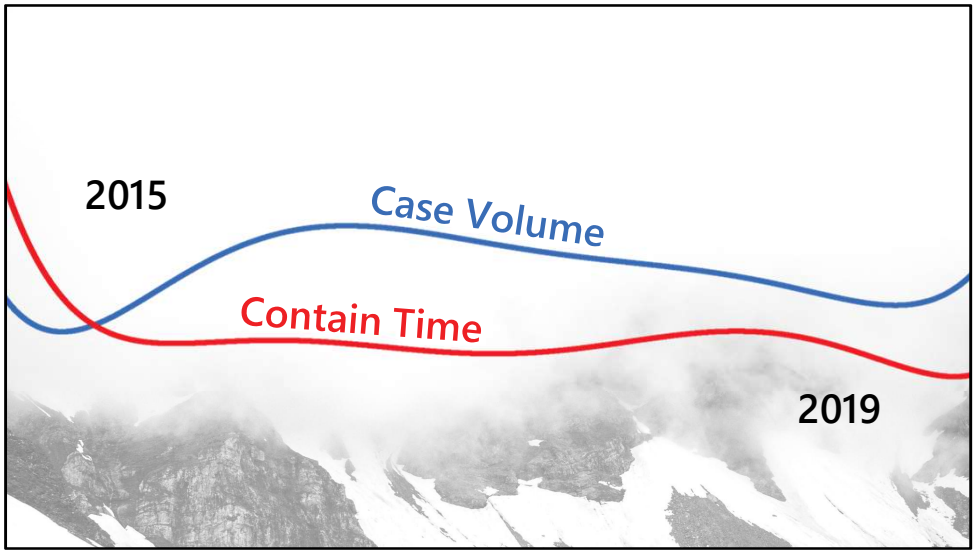
A team we call Continuous Improvement is dedicated to security measurement and accountability. This team tracks security performance across multiple groups, and is the primary facilitator in the Red – Blue relationship.

After each operation, the data that both Red Team and Blue Team generate are compiled and compared in an after-action report. Because this report is created by a third party, we have the benefit of a more objective viewpoint.

Then that report is reviewed in a meeting with members from all of our partner teams. There's usually twenty or so people in the room for this.

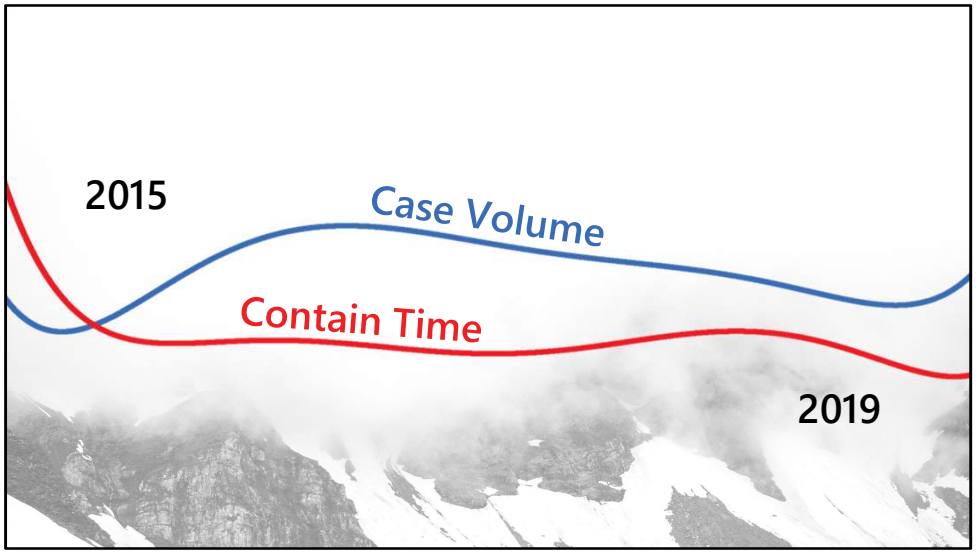


The entire timeline of the operation will be reassembled from the available information, and everyone in the room during these debriefs will look for opportunities to improve.



Continuous Improvement also sifts through and reports on a large amount of security data. For example, an analysis of four years' worth of case information.

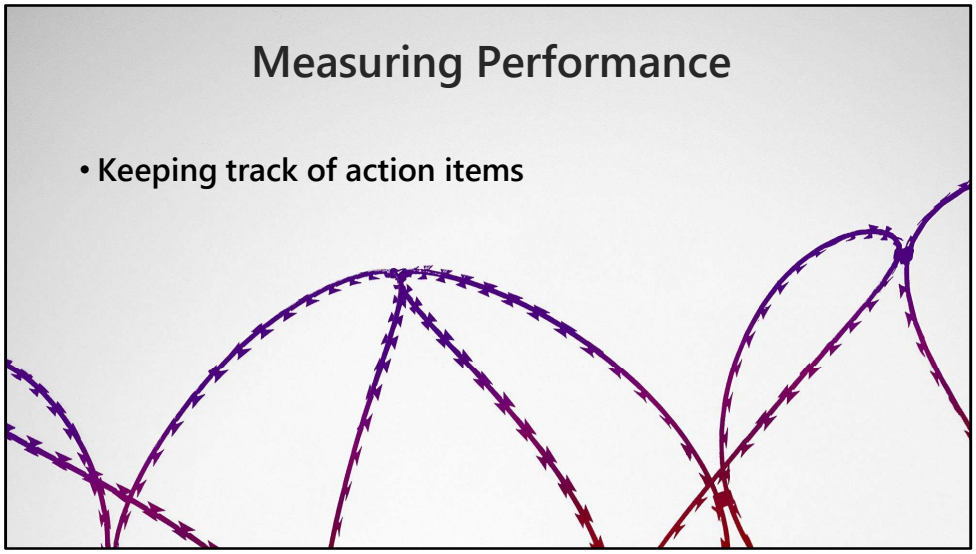
This chart illustrates how the weekly volume of malicious cases correlates to the average time it took to contain the threats.



This suggests that there are causes other than volume which impact contain time, since there appears to be little direct correlation.

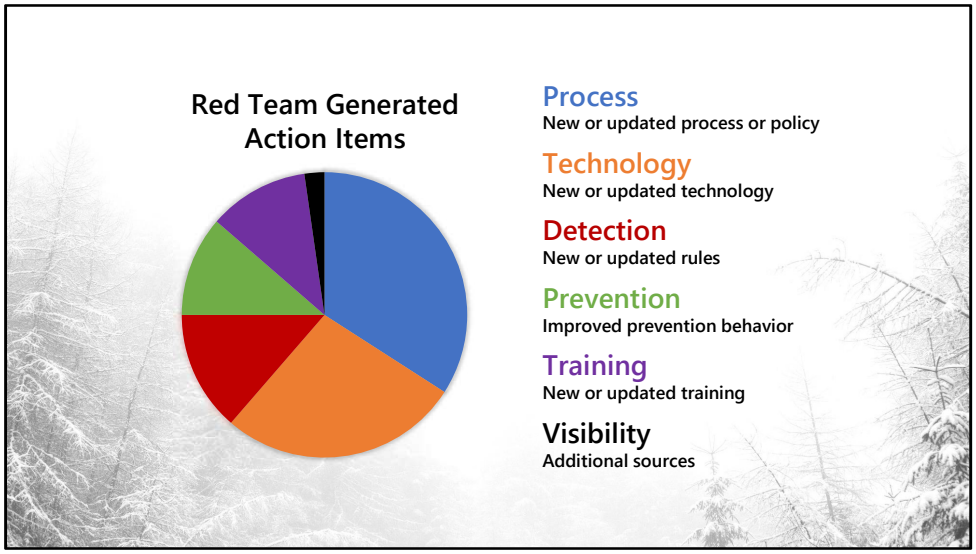
Continuous Improvement also regularly meets with management to discuss overall security performance.

Red Team backed data is a strong contributor to those discussions.



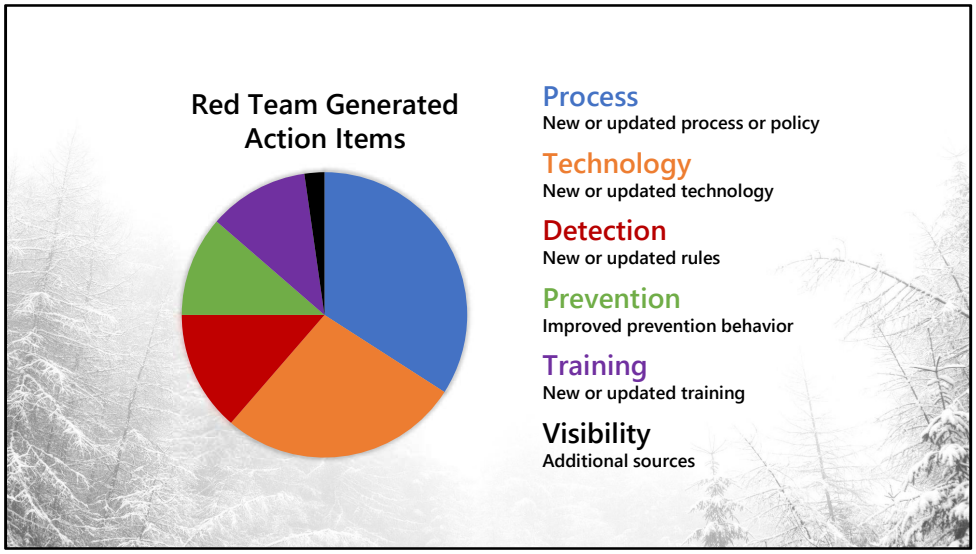
We've found only a few ways to quantify Red Team performance. The most important to us seems to be the number of Red Team generated action items.

We estimate as long as Red Team is identifying areas for improvement, and those findings are being addressed, we are hitting our performance benchmarks.



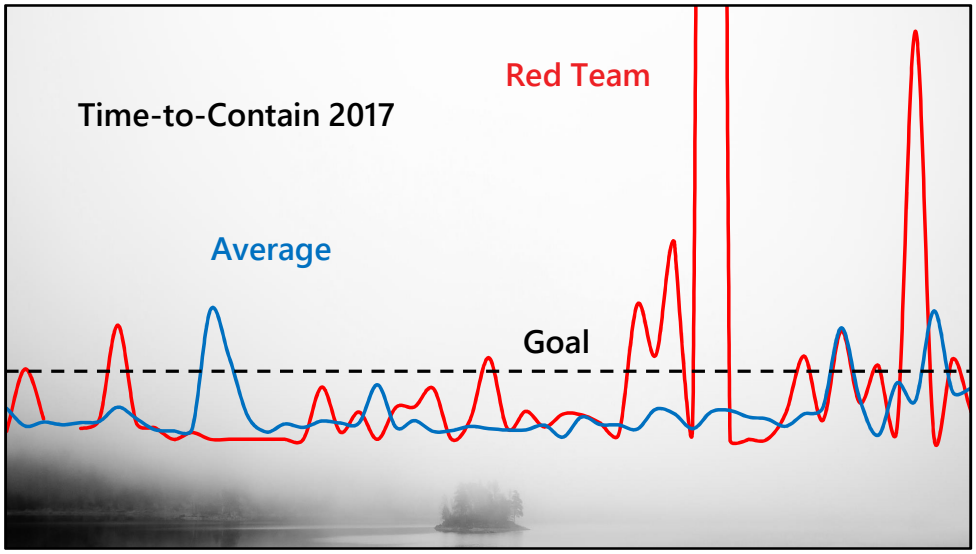
One of the metrics Continuous Improvement tracks is where each action item lands in terms of how the action item might be addressed.

This is a breakdown, over a four month period, of all the action items our Red Team generated.



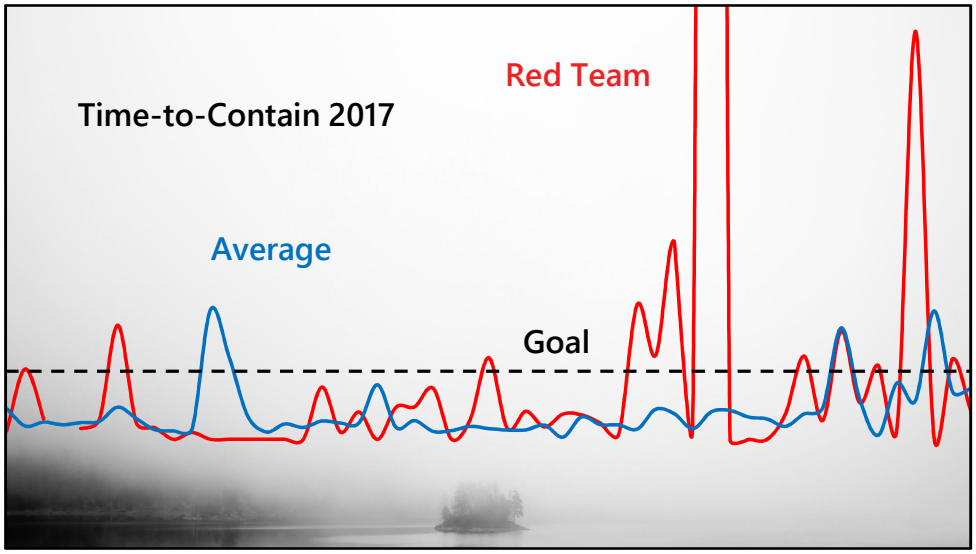
Some time ago, visibility would have represented a larger cross-section of the action items.

As our Red Team spent more time operating in areas where better visibility was needed, those gaps were reduced. We're making sure our Red Team is addressing the areas most in need of improvement.



We can also examine Red Team performance using our containment time metrics.

This is data for an entire year. We can see that containing our Red Team often takes longer than the average malicious case, but we're rarely too far ahead for too long. Tracking this data helps Red Team tune its operations to match Blue Team capability.



We are maturing along with Blue Team, not running far out ahead where the dragons of diminishing returns live.

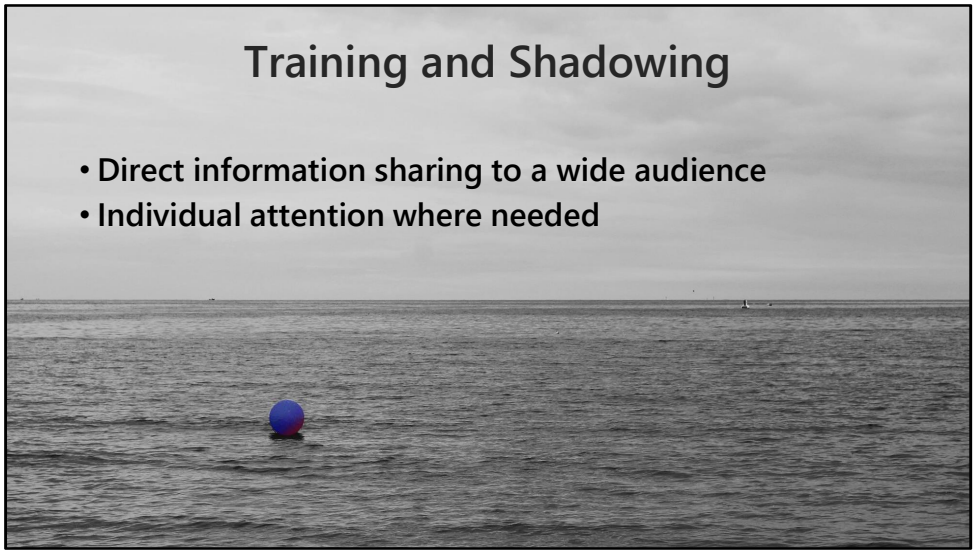
Another way to measure Red Team is frequency of operations meeting established goals.

There's rarely a shortfall for us here, as our goals are generally simple, and aren't all-or-nothing. If there are unexpected results in an operation, there's usually a Plan B, or Plan C, which will also satisfy operational goals.

Allowing for graceful failure is one of the ways we maintain a positive working environment, and assure that WEEKS or MONTHS of work rarely goes to waste because of a SINGLE mistake.

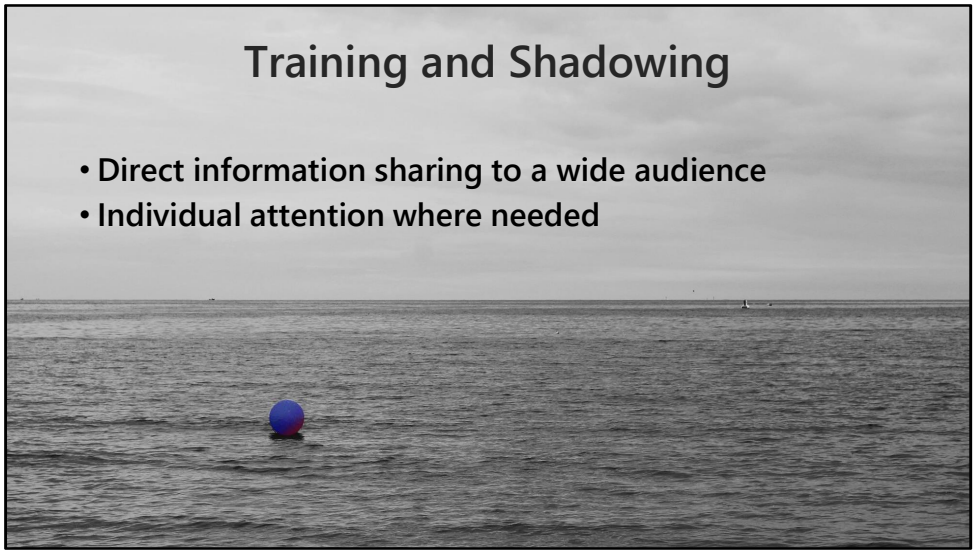
If we can provide some consistent operational data, performance measurement is easier. If something, somewhere was learned by someone, we consider that success.

If not, as we say on Red Team, there's always next week.

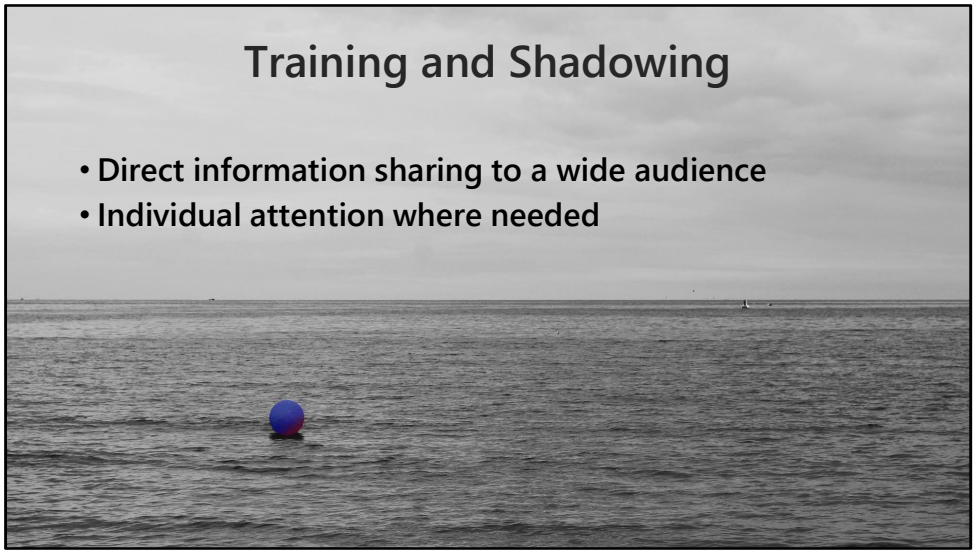


Another way to gauge our Red Team performance is how much mentoring, training, and professional development happens. As long as our Red Team is continuously learning, and passing that knowledge on to other teams, and junior team members, we will remain high-performing.

In service to this, our Red Team performs direct training.



We know that if our partners understand how we do business as a Red Team, they're more capable of communicating their needs, and helping us meet ours. We present on various topics to large groups, like reverse engineering, and malware development.

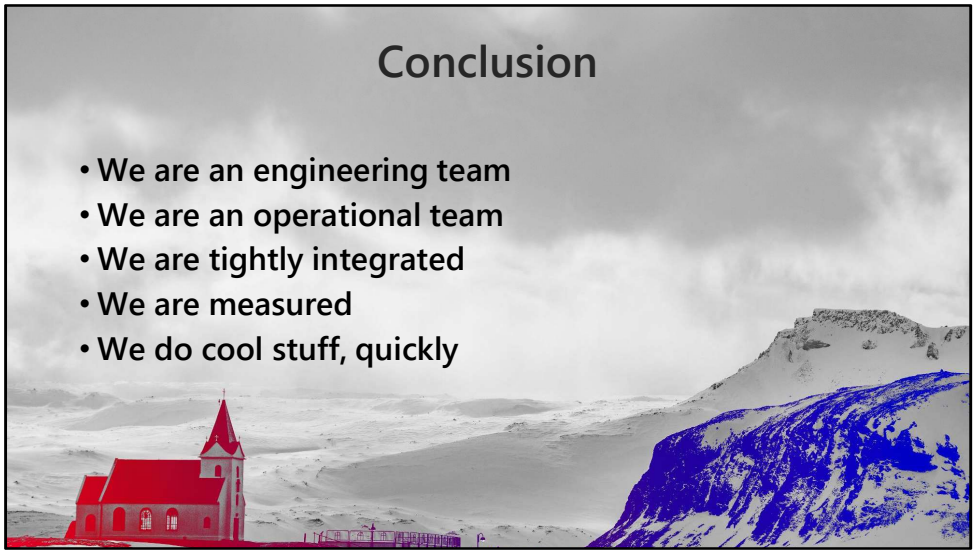


We do debriefs directly with analysts, who have expressed that one-on-one time with our Red Team is important to them. They get the opportunity to ask questions in a safe environment.

Our team, like others in the security organization, operates a shadowing program. Members from other teams are spending a few days working alongside our Red Team, and getting some whiteboard theory on how we work.

As well, our Red Team shadows other teams, giving us the opportunity to get a much better understanding of how the organization as a whole functions.

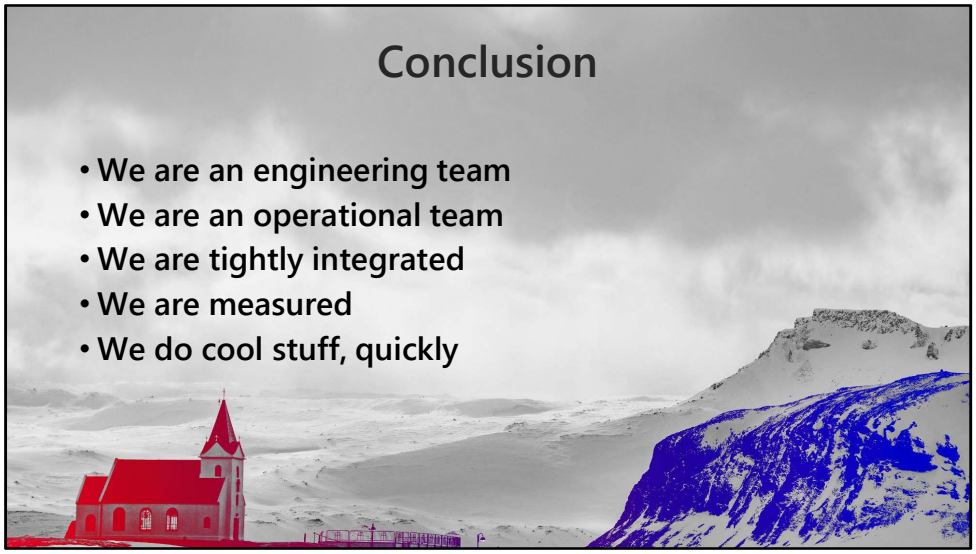
This process allows us to be empathetic about how we do our work, and get the best value for every hour of every day we put into it.



We've become a proper software engineering team, capable of planning, developing, and delivering stable products. We deliver services based on those products.

We're tightly integrated with other security teams, with great working relationships.

We're capable of measuring what we do, and using that to improve week-to-week.



We can do really cool things, and do them quickly.

It's been years, and we're still at. We've done it with as few as four engineers. We do it far better with a team twice that size.

Since we've become more strategic about what, when, and how we do operations, we're all happier, and have drastically reduced the amount of labor required to do quality work.

There's no one right way to do Red Team. I just hope our story will inspire you to try new things, that, for us, have driven meaningful change.