LEADERSHIP FOR CYBER AND TECH PROFESSIONALS

Based on "How to Win Friends and Influence People" by Dale Carnegie
and "The Leadership Challenge" by James Kouzes and Barry Posner

1. INFLUENCE WITHOUT AUTHORITY

Most cyber professionals can't force anyone to patch, can't make a developer fix their code, can't stop a project from shipping, and still get blamed when something breaks.

What this looks like in practice

- Getting an engineer to prioritize your vulnerability when they have 40 other tickets
- Convincing a team to change a config without escalating to their manager
- Making security feel like part of their job, not extra work you're dumping on them
- Asking "what would make this easier for you" instead of "this is due Friday"

The moment you rely on authority or escalation, you've already lost ground.

2. MODEL THE WAY

People follow what you do, not what you say. The manager who demands patching but ignores vulnerabilities on their own test box. The lead who requires documentation but never writes any. The architect who mandates code review but pushes straight to prod "just this once."

What this looks like in practice

- You use the same MFA, VPN, and access controls you enforce on others
- When you mess up a firewall rule or miss something in a review, you say it out loud
- When your team ships a bug, you stand in front of leadership, not behind your team
- You never say "who approved this" when you already know the answer

People remember hypocrisy forever. They also remember when you held yourself to the same standard.

3. MAKE PEOPLE FEEL IMPORTANT

This feels soft. It is not. Engineers get blamed for vulnerabilities they inherited. Ops teams get woken up at 2 AM for problems they didn't create. Developers hear "security" and brace for criticism.

What this looks like in practice

- In a meeting, someone catches a misconfiguration. You say "good catch, thank you" out loud. Takes three seconds.
- Someone makes an error. You Slack them privately instead of replying all or calling it out in standup.
- Before asking a team to change something, you acknowledge they're already underwater with other priorities.
- You treat the developer who doesn't know what a CVE is as an intelligent person who just hasn't learned this yet.

If people feel embarrassed or dismissed, they will resist you. Sometimes consciously. Sometimes not.

4. INSPIRE A SHARED VISION

Cyber leaders fail when they lead with control language, fear, or compliance framing. They succeed when they connect security to something people actually care about.

What actually works

- "Being diligent with audit prep now means we're not scrambling in December when we'd rather be with family."
- "If we fix this in design, the on call engineer doesn't get paged at 3 AM six months from now."
- "Getting this right means the product team ships faster, and we look like the team that enabled it."

- You make people feel like they're protecting something that matters, not just checking a box.

People don't rally around CVEs. They don't rally around compliance. They rally around purpose.

5. SEEK FIRST TO UNDERSTAND

Security solutions fail when you don't understand the system you're securing. Resistance usually isn't laziness. It's hidden risk, technical debt, or workload you didn't know about.

What this looks like in practice

- Before recommending a fix, you ask "what's getting in the way of this for you?"
- Before pushing a new tool, you ask "what have you tried before?"
- Before escalating, you ask "what would break if we did this?"
- You find out the team already tried your solution two years ago and it caused an outage.

Listening first isn't slow. It's how you avoid wasting everyone's time, including yours.

6. CHALLENGE THE PROCESS

Cyber leaders must challenge the architecture that made sense five years ago, the default config that shipped insecure, the "we've always done it this way" that nobody has questioned. But you challenge systems, not people.

What this looks like in practice

- Instead of "this design is insecure," you say "what are we trying to protect here, and does this approach get us there?"
- Instead of "this is wrong," you say "what would it take to test a different approach?"
- Instead of "no," you say "here's what we'd need to avoid" and let them problem solve with you.
- You propose experiments, not mandates.

The goal is progress, not purity. You're not here to be right. You're here to make things better.

7. AVOID CRITICISM, CONDEMNATION, AND BLAME

Post incident behavior defines your reputation permanently. If you've ever sat in a review where leadership asked "who approved this?" you've seen the room go silent. That silence is expensive.

What good looks like

- You ask "what made this easy to miss?" instead of "who missed this?"
- You focus on the system that allowed the error, not the person who made it.
- You treat the incident review as a learning exercise, not a trial.
- You make it safe to say "I made a mistake" without career consequences.

Teams who feel safe report issues earlier. People who fear blame hide problems until they explode.

8. ENCOURAGE THE HEART

Cyber teams live in constant urgency, invisible success, and thankless prevention work. Recognition isn't fluff. It's fuel.

What this looks like in practice

- You thank someone for the boring, correct work that prevented an incident nobody will ever hear about.
- You acknowledge effort when a project fails, not just outcomes when it succeeds.
- You celebrate the avoided incident, not just the heroic recovery.
- Once a week, send one message thanking someone for something specific. Not "great job." Something like "thank you for catching that config issue before it hit prod."

Burnout kills more security programs than attackers do. Recognition is one of the cheapest tools you have.

LEADERSHIP ACCORDING TO ME: FAILURES AND FREEBIES

- Be gracious when able. Crushing someone's spirit who had good intent is a trust killer.
- You still have to produce. Showing you do work matters.
- Toxic environments need different leadership. Sometimes your whole year is boundary setting or trust rebuilding.
- Look at your processes, people, and projects from a birds eye view.
- Be the enabler, the tactful honest professional, the one who listens first, protects their team, and remembers that people rally around purpose, not policies.