keystrike

Establishing the root of trust in the age of APT

How Identity Controls Conflate Devices, Accounts & People

Written by Andrew Fife

The circular logic in zero trust principles

We assume the device is compromised and yet we store authentication tickets, tokens, cookies, keys and credentials on the device that is... compromised?

MFA was designed to address the issue of weak, stolen, and leaked passwords. PAM was designed to protect admin account hijacking. However, in practice, MFA and PAM have become the cornerstone of protecting our most sensitive systems – Something that they were never intended for.

Why? We all know that identity is the foundation of security. Authorization is literally the difference between a breach and appropriate use, and authentication is how authorization is enforced. Hence, strong authentication is supposed to confirm that the user requesting access is the person they claim to be, but, in the most important scenario, this is precisely where authentication fails.

Instead of securing the network, MFA and PAM taught attackers how to be even more stealthy. Attackers and defenders are inherently locked in an evolutionary relationship:

01 The rise of SaaS increased incentives for attackers to compromise passwords.

02 Defenders responded with the widespread adoption of MFA.

03 MFA forced attackers to wait until after authentication to target access tokens, often by compromising workstations.

04 Defenders then protected access tokens on workstations with PAM to ensure that privileged access tokens for sensitive systems were only valid for short time windows.

05 Forced to attack during short time windows, attackers built triggers to detect when remote sessions are established to launch attacks from the workstations they compromised.

The result is that defenders identity controls have been bypassed and, because MFA and PAM forced the attack to come from the authorized workstation while legitimate tasks are being performed, the primary signals for network and behavioral anomaly detection are simply gone.

Flaws in Device-Centrics Identity Controls

The fatal mistake in device-centric identity controls is conflating the device and account with the authorized person. Today, breach after breach demonstrates attackers executing same pattern:

01 Gain an initial foothold by compromising a device via social engineering.

02 Hijack credentials, keys, tickets, tokens, etc. stored on the device.

03 Use the authorized account to move laterally to another device.

04 Rinse and repeat until compromising a device with access to a sensitive system.

05 Deploy ransomware and/or exfiltrate critical data.

Advanced persistent threats (APT) typically move laterally by impersonating authorized users in two ways. First, they learn the users' behavior patterns and then they take over the user's accounts. Since attackers have access to the authentication tokens, API keys, passwords, decryption keys, and so forth stored on the device, after compromising a device, injecting malicious commands into active sessions is trivial.

Identity security layers that pin access to devices inadvertently assume the device is not compromised by conflating the identity of the device with the authorized person. In fact, this conflation is so ingrained that, in practice, MFA is the cornerstone of Zero Trust Network Access (ZTNA) implementations. This makes no sense if the core tenet of "zero trust" is: Assume every device is compromised. Yet,after the OTP is gleaned from a 2nd factor, an MFA token is stored on the device... that we assume is compromised?! Clearly it is circular logic to rely on MFA for the root of trust in ZTNA.

Authentication is a Point in Time

One of the biggest assumption flaws in authentication is that anything that occurs after the initial login is not malicious. The login is temporal but for convenience most authenticated sessions are persistent. Even well thought out services, like Google Workplace, default to weekly session timeouts when many attack techniques only require a few minutes. However, even tight session timeouts don't slow down APT either because attackers can simply work in parallel with the authorized person via process injection. To block lateral movement and secure the full session, we must determine the intent of every command so that we know which came from the attacker and which came from the authorized person.

The Swiss Cheese Model of Defense in Depth

Lining up slices of swiss cheese illustrates a realistic approach to creating independent layers of security that add up to defense-in-depth:

- Each security control represents a slice of swiss cheese.
- No security control is perfect and all of them can be bypassed just as each slice of cheese has holes in it.

The key goal of the model is collecting and aligning security controls such that they cover each others' weaknesses, just as if swiss cheese slices were lined up such that no single hole breached all slices. The goal of any defense in depth strategy is to leave no scenario in which the attacker can defeat all security controls at the same time.

The opposite of security layers are when each control suffers from the same achilles heel. Which is exactly how we protect sensitive systems today:

Network Segmentation: Kerberos ticket on the device.

Single Sign-On: SAML tokens on the device.

MFA: MFA tokens on the device.

PAM: JWT tokens on the device.

Encryption: Decryption keys on the device (or via enterprise vault).

On the surface these controls look like independent layers, but in reality they all suffer from the same achilles heel: When the device is compromised, the attacker can reuse these tokens, keys, and sessions as easily as the authorized user. Since device compromise is no longer rare, it's time to rethink how we enforce identity and validate that the commands executed by the system are generated by the authorized user and only the authorized user.

But I have short timeout windows...

Unfortunately, short timeout windows don't offer material security from APT who've compromised workstations. Authorized people legitimately using their accounts and attackers hijacking access don't have to work sequentially. Successful attackers often leave behind scripts that wait for valid sessions to open and inject their malicious commands along with the legitimate users' commands. By traditional security attributes (network, behavioral, etc.), the commands are difficult to distinguish and executed together.

But I have PAM...

When a device is compromised, Privileged Access Management (PAM) and Just-in-Time PAM (JIT-PAM) do create an extra hurdle for attackers to clear, it just isn't a very difficult one. Programmatically, it's not difficult to determine when new sessions have been created. Attackers build triggers into their malwareware to wait for JIT PAM to deliver tokens to the device, and then work in parallel with the authorized person. The bar is higher with PAM but it still fundamentally conflates device identity with the authorized person's identity.

But I have Cheddar Cheese!

If identity-based security controls are not as resilient as they seem, our non- identity controls should create meaningfully different layers of defense. However, when the device is compromised these more differentiated controls are of little use.

- **Anomaly Detection with EDR or IPS:** Good attackers don't do obvious things like port scanning the entire network or conduct their attacks during off hours. Effective attackers live off the land to learn behavior patterns and there's no anomaly to catch when an attacker is using an authorized account at the same authenticated session as the authorized person.
- **Orchestration & Automation:** This requires both knowledge of a pattern and a pattern to occur. If we can't identify that the device is compromised, and behavioral patterns are inaccurate, there's no reliable signal to trigger automated workflows.
- **SIEM:** EDRs and IDS dumping mountains of alerts on the SOC, SIEMs do help analysts identify breaches, but only once they identify enough data points to draw the right conclusion. This could require waiting for activity over days or weeks while a savvy attacker is deploying their attack in minutes.
- **User training:** The holy grail! Drinks on me if you figure it out!

This is not to say that these tools don't have value. All of them have important roles to play, they just tend to help piece together what happened after the breach has occurred as forensic tools rather than blocking attackers' moving laterally in real-time to actually stop breaches.

Assumming APT

To build security controls that work when the device is compromised, which is when security is needed most, we must reframe our thinking. If we assume an APT has compromised a device then we must include the characteristics of how APT operates in our threat models:

01 Operating with long time-horizons

02 Seeking to impersonate authorized users

03 Adapting quickly

04 Learning behavior patterns

05 Detecting when new sessions are established

06 Building triggers that time attacks perfectly...

Do we really have any effective security controls at all? Daily breach headlines suggest that the answer is no.

Breaking The Device-Centric Identity Paradigm

As technology evolves, the first wave of solutions are attempting to address the last threats, not adapting to the changing attack profiles. The impact of that change is grossly underestimated. It means that extending pre- existing approaches to use the old tools to address the new threats will typically miss the mark. This is exactly what has occurred with MFA, which was designed to protect against compromised passwords, not modern APT.

As breaches have shifted to APT that live off the land, we've tried to adapt MFA to be the core of the ZTNA model but this underestimates the real change occurring: It is easier than ever for attackers to mimic authorized users. To solve the new problem we must find ways to enforce the ZTNA philosophy without falling prey to the circular logic of relying on device-centric identity controls.

Intent-Based Authentication

Effective security in the face of device compromise requires full-session security that doesn't just authenticate connections. Continuous authentication, and adaptive authentication extend pre-existing approaches. They solve the temporal nature of authentication, but they assume all commands in authenticated sessions are legitimate.

In order to stop APT we must assume the machine is compromised and therefore the session is compromised and therefore every command is suspect until proven otherwise. If the goal is to stop breaches, then we must detect malicious commands with high enough accuracy to automate decisive action to block in real time. If we have to send an alert into a SIEM and wait for correlation, it's already too late. In order to block APT moving laterally today, security controls must:

- Remain effective even if the device is compromised.
- Protect the entire session, not just the initial login.
- Distinguish between legitimate and malicious commands.
- Isolate sensitive systems from malicious commands in real time.
- Avoid disruption to productivity and MFA fatigue.

The Way Forward: Authenticate Commands, not Connections

The array of device-centric identity controls, including MFA and PAM, is falling short in defending against the larger threat. Device-based, and continuous security tools are relying on a flawed assumption that devices remain uncompromised—a stance that breaks down against real-world attack strategies.

Attackers capitalize on this breakdown moving laterally within networks by mimicking user behaviors and leveraging compromised credentials. Short timeout windows, MFA, and even privileged access management (PAM) ultimately share the same weakness: they assume device identity is a proxy for the human behind it. We've seen how that is not a safe, or correct assumption.

Intent-based authentication changes the prevention model by verifying every command throughout the session. Rather than relying on static, periodic, or even continuous checks, this approach uses real-time assessments of each command's intent to close the gaps that device-centric methods leave open.

This means Intent-based security remains effective even when devices are compromised, providing session-wide protection that guards against lateral movement and advanced persistent threats (APTs). It's clear that identity verification needs to go far beyond device-bound assumptions. Intent-based authentication reframes identity in a way that aligns with zero trust principles, adapting to behavioral signals that attackers cannot learn or emulate.

For organizations committed to effective, forward-looking security, intent-based authentication is closing the gap. Now user intent, not device identity, serves as the foundation of defense and a reliable root of trust. In a world of ransomware-as-a-service, increasingly every organization with critical data must become forward looking and adapt to the world of APT living off the land.

keystrike

www.keystrike.com info@keystrike.com