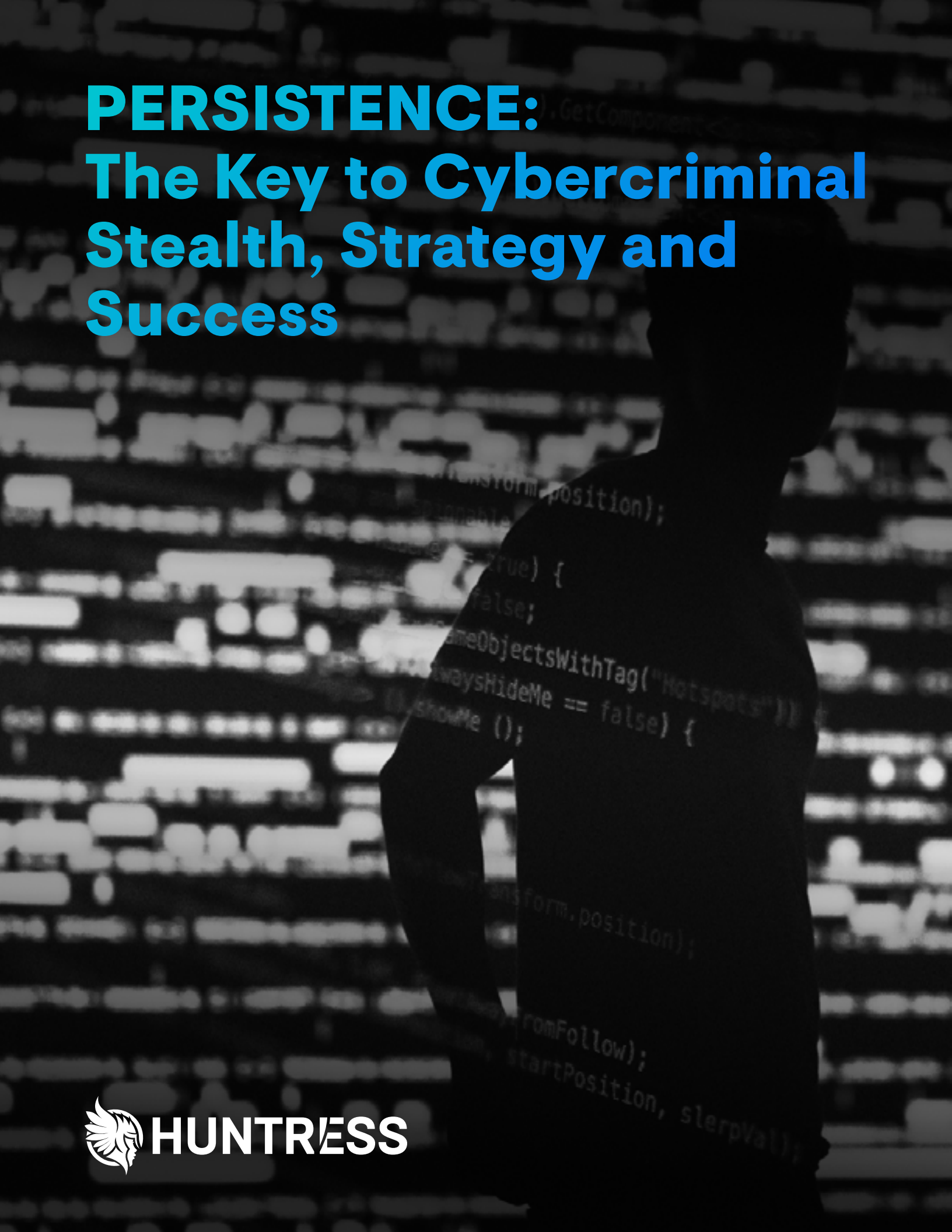
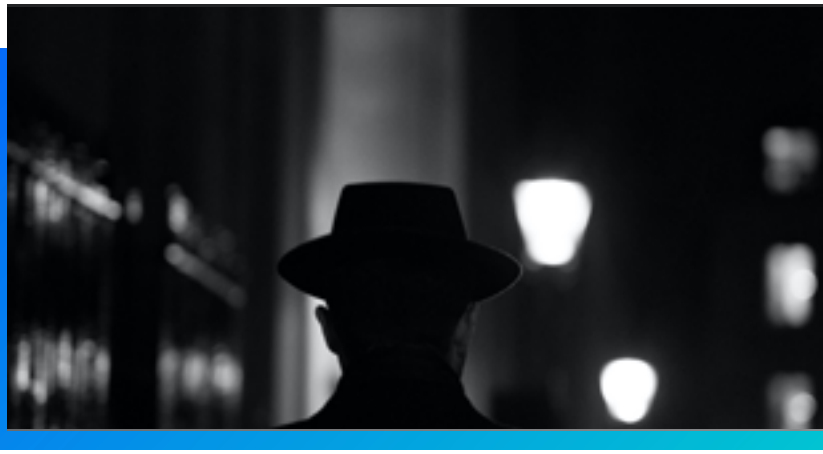


PERSISTENCE: The Key to Cybercriminal Stealth, Strategy and Success

HUNTRESS

Table of Contents

In today's game of cybersecurity, both the rules and the players have drastically changed.



The warning signs of a cyberattack aren't as glaring or obvious as they once were. The risks and damages are worsening. We are now defending against a more organized and formidable force—one that hides in our blindspots and outpaces the defenses many businesses have in place.

What makes these modern attackers so dangerous? For starters, they're acting more like legitimate enterprises than they are criminal organizations. They've got product teams, sales staff and entire marketplaces on the dark web where they're conducting business. And just as the business of cybercrime has continued to evolve, so has the tradecraft bad actors are using to infiltrate their targets.

Instead of writing a virus that spreads sporadically, today's attackers are exploiting native parts of an operating system to evade detection and hide in plain sight. Their attacks are designed for stealth and sneaking past prevention—leaving many security tools incapable of stopping them.

In this eBook, we'll highlight why persistence has quickly become a staple in the modern attacker's playbook. We'll also get our hands on the keyboard to showcase exactly how hackers use persistence to hide—and how you can flip the script and use it to seek them out.

The Problem with Prevention

Perfect prevention doesn't exist. When we think about prevention-based technologies—firewalls, antivirus, multi-factor authentication, etc.—they're designed to block attackers from gaining access to a targeted environment. They act like moats around a castle, preventing attacks by defending the perimeter. And while a critical defense mechanism, they shouldn't be the only line of defense.

Like moats, security strategies that focus entirely on prevention are a bit antiquated. These traditional methods rely too heavily on signature-based detections, automation or machine learning to block what is *known* to be bad or malicious. But what about threats that disguise as something benign or "normal"?

Always evolving their tradecraft, hackers are getting better at outsmarting these preventive defenses. They know they can't phish the same victim or exploit the same "loud" vulnerability repeatedly—that type of overt behavior will get them caught. Instead, they've become masters of disguise.

Attackers have figured out ways to slip through outer defenses undetected, often hiding within legitimate applications and processes. And once inside, their goal is to set up camp and dwell on a device while they plot their next move. But first, they need assurance. After all the effort of sneaking in, they want to ensure they don't lose their hard-earned access. **That's where one of today's more innovative—and pesky—attacker techniques comes into play: persistence.**

What Is Persistence?

Persistence is like a piece of tape placed on the latch of a door to prevent it from locking. It's an attack tactic used to discreetly maintain long-term access to systems across restarts, changed credentials or other interruptions that could cut off a hacker's ingress.

Persistence is used for its subtlety and stealth. Attackers typically create persistence mechanisms, what we call footholds, by exploiting built-in functionality of an operating system—allowing them to both bypass preventive tools and remain hidden until they are ready to make their next move.

> **Persistence is like a piece of tape placed on the latch of a door to prevent it from locking. It's an attack tactic used to discreetly maintain long-term access to systems across restarts, changed credentials or other interruptions that could cut off a hacker's ingress.**

WHY PERSISTENCE?

Let's step into a hacker's shoes for a second. You've just spent all this time and effort gaining initial access and sneaking into your target's system—now imagine losing all of that to a simple reboot of the machine.

Persistence provides hackers with a longer shelf life and easy re-entry should they need it. Essentially, it's an attacker's safety net. It adds a hint of certainty into an otherwise uncertain process.

And not only does persistence make an attacker's life easier, it's also hard to catch using traditional tools and methods. Automated security software might assume a given file is defunct or dormant, when in reality, it's a foothold established by a bad actor. Most preventive tools won't pick up on anything suspicious—and that's exactly what attackers are banking on.

PERSISTENCE-ENABLED ATTACKS VS. OTHER THREATS

Persistence is often thought of in the context of advanced persistent threats (APTs). While APT attacks are certainly a type of persistence-enabled attack, not all persistence-enabled attacks are APTs.

In the last few years, the lines have blurred between the attack capabilities of nation-state players and those of smaller cybercrime groups. The techniques and tools characterized by a few APT actors have been distilled down and adopted by hundreds of lower-level threat actors. This means that the tactics once used for more high-profile and large-scale attacks—like persistence—are now becoming more common among smaller and less-prominent companies.

> The tactics once used for more high-profile and large-scale attacks—like persistence—are now becoming more common among smaller and less-prominent companies.

Attacks that use persistence are particularly dangerous because, by nature, they are difficult to detect. These attacks rely on their ability to fly under the radar, all the while remaining out of sight of traditional security measures. And they always tend to have tricks up their sleeve—even when the threat is discovered and the immediate danger seems to be gone, hackers might slip back in through another secret backdoor.

A key differentiation in persistence-enabled attacks is their "low and slow" approach. We've become accustomed to the typical cyberattack being like the hare: adversaries act quickly and carelessly to achieve their mission. In contrast, persistent threat actors are more like the tortoise: their steady and methodical manner is key to keeping a low profile while they carry out their attack.

Although slower, this persistent strategy enables more stealthy—and ultimately more successful—exploits.

WHERE DOES PERSISTENCE FALL IN AN ATTACKER'S WORKFLOW?

Establishing persistence is an attacker's top priority after initial access. Once they've scoped out their target and found a way in, they'll leave the back door open so they can slip in unannounced at any time. After all, it might take more than one visit to accomplish their goals.

Persistence has quickly become a staple in the modern attacker's playbook, and it typically falls right in the middle of their workflow. In the beginning stages, persistence is strategically used to blend in with the background after they've snuck in. Towards the end, it's used to avoid getting caught by the user, IT staff or security software while they plan the next stages of their attack.

1. Reconnaissance

Goal: Gather as much information as they can.

This step is critical in solidifying an attack's "mission." Any information gathered—whether it's specific vulnerabilities to exploit or users to phish—can be leveraged by the adversary to aid in other phases of their workflow.

2. Initial Access

Goal: Find a way in.

During this phase, hackers will do anything they can to gain unauthorized access to their target's system. The method chosen here often reflects more on the skills of the attacker than the weaknesses of the target, but common techniques include social engineering, website hacking or vulnerability exploitation.

3. Persistence

Goal: Stealthily maintain access without getting caught.

This step is all about establishing and concealing their presence. Techniques used for persistence include any access, action or configuration changes that let an attacker maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code. Not only does this buy them more dwell time without raising any red flags, it also allows them to hide the intrusion long after they have left.

4. Discovery

Goal: Get a lay of the land.

Hackers use this phase to gain knowledge about their target's system and internal network. Adversaries will typically explore what they can control within the environment and what's around their entrypoint in order to discover how it could benefit their current mission.

5. Execution

Goal: Make their malicious move.

At this point, it's time to set the plan in motion. This execution stage can take many forms—it all depends on the initial mission, the skill level of the hacker or what they've discovered along the way. The outcomes here could be anything from data exfiltration, dropping ransomware, mining cryptocurrency, vandalizing a website, or even selling their access or stolen credentials. Whatever the motive, it's usually malicious.

The stealth and success of an attack hinges on persistence—and the key to persistence is to not be detected.

Hackers have near-perfected the art of evasion. With the right persistence mechanisms, they're able to lurk in the shadows for extended periods of time. In fact, M-Trends' 2021 Report found that the median dwell time an attacker is present in a victim environment before they are detected is 24 days. That's ample time to lay the foundation for a stealthy cyberattack—and most victims are none the wiser.

HOW HACKERS EVADE DETECTION WITH PERSISTENCE

Unlike ransomware or denial of service attacks, you can't see persistent threats right away. They're designed to hide in plain sight—abusing legitimate applications and processes to evade being detected by antivirus or other preventive security measures. Some persistent threats are so skilled at this that they can even bypass more than one preventive product.

Of the security incidents that Huntress detected from January to May 2021, 73% of the persistent threats we saw were on endpoints where one other cybersecurity product was installed. What's more concerning is that 19% of these incidents had two other products installed, and yet persistent threats were still able to get through. This is why we talk about the importance of *layered* cybersecurity—even with multiple security measures in place, attackers are still successfully evading these outer layers and lurking in the shadows.

73% of persistent threats were found on endpoints with **one** preventive product present.

19% of persistent threats were found on endpoints with **two** preventive products present.

Source: Data captured from the Huntress Security Platform, which detects persistent footholds and other cyberthreats, January–May, 2021 (Base = 2,566)

COMMON PERSISTENCE TECHNIQUES

MITRE ATT&CK®, the gold standard for understanding adversary tactics and techniques, lists 19 different **known** techniques attackers use to achieve persistence, each with their own set of sub-techniques.

Some of these techniques are very broad, some are extremely narrow, and others likely exist that we're not yet aware of. But from what we do know, here are the most common persistence techniques we see being used out in the wild:

Boot or Logon Autostart Execution

Operating systems have various mechanisms for automatically running a program on system boot or account logon. This is a native function that many attackers abuse. They can maintain persistence on compromised systems by configuring settings to automatically execute a program during system boot or logon.

One of the more common ways attackers achieve this is by adding an entry to the "run keys" in Windows Registry or Startup folder. This will cause any referenced programs to be executed when a user logs in.

This technique relies on abusing system features, which makes it difficult to flag and mitigate using solely preventive measures.

Boot or Logon Initialization Scripts

Similar to the above technique, hackers can also use scripts that are automatically executed at boot or logon to establish persistence. Initialization scripts can typically be used to perform administrative functions, which means it can give attackers an ability to execute other programs or send information to an internal logging server.

Depending on the access configuration of the logon scripts, either local credentials or an administrator account may be necessary for this technique. Ensuring proper permissions and restricting write access to logon scripts to specific administrators will help keep risk down—but not down to zero.

Scheduled Task/Job

This technique involves abusing the task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time.

A common method is abusing Windows Task Scheduler, which can be used to execute programs at system startup or on a scheduled basis. As an example, TrickBot, a trojan spyware program, has been known to create scheduled tasks on compromised systems in a way that provides persistence for the attack.

This is a tricky one to distinguish because legitimate scheduled tasks may be created during new software installations or through system admin functions—so it's worth keeping an eye out for changes to tasks that do not correlate with known software, patch cycles, etc.

These are just a few common examples—there are many different ways an attacker can go about establishing and maintaining persistence. And more often than not, persistence is a key indicator that an adversary has already slipped past preventive defenses and successfully gained initial access.

WHAT DOES PERSISTENCE LOOK LIKE?

Let's say, for example, an attacker is able to compromise a system and create a scheduled task that automatically executes the following command every time the machine starts up:

```
cmd /c "start /b
```

This kicks off a new command prompt in the background.

```
c:\ProgramData\48756e74.bat"
```

This is the location of the batch file to be executed.

At a glance, it is easy to focus on the second half of this command; there is clearly a very unusual-looking file being called. Let's go ahead and open the file to see what's inside:

```
net user eviluser "myEvilPassword" /ADD
net localgroup administrators eviluser /ADD
```

This batch file adds a new backdoor account with administrative privileges.

In this case, the challenge an automated security tool would have is validating malicious intent with this scheduled task—and that's to the benefit of the attacker.

Many preventive tools require a high degree of confidence that malicious activity is occurring before stepping in. Creating a username and password through a command line prompt could actually be a legitimate administrative task. Therefore, most security products will allow the action to continue in order to avoid potential disruption for the end user. All the while, the attacker can stealthily hide in the software's blindspots.

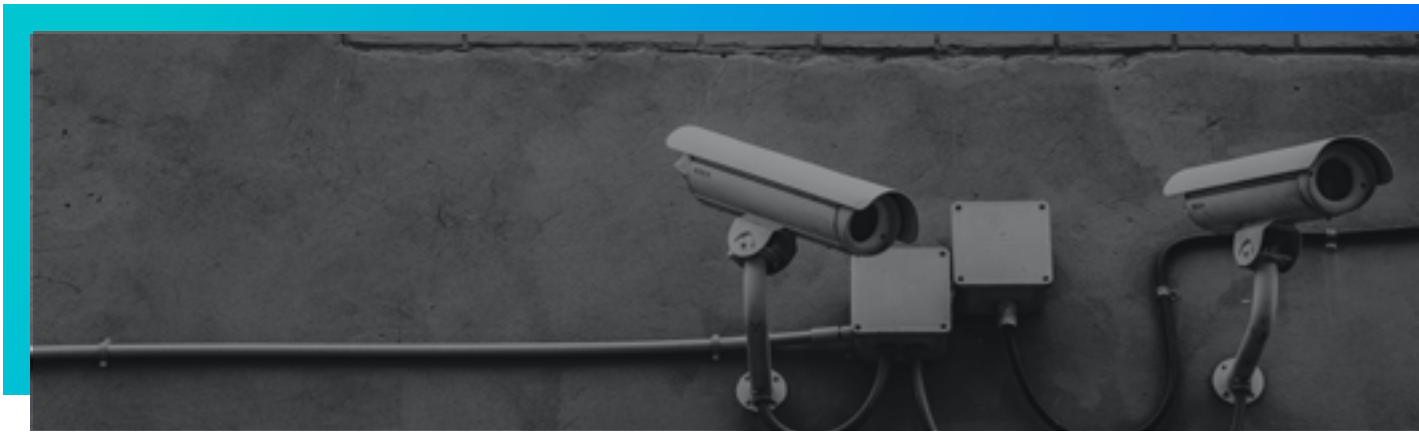
This is why persistence is an attacker's greatest ally. It provides secret, backdoor access that's hidden within the existing parts of an operating system. And while extremely useful to bad actors, persistence can also be the smoking gun at the scene of the crime.

How to Hunt for Persistence

If you were to discover a piece of malware on an endpoint and just delete it, there's a good chance it will find a way right back in. This is because you're only treating one symptom, not the root problem. Addressing that root problem—the persistence—is essential in short-circuiting an attacker's workflow. Finding persistence allows you to uncover the rest of the malware, flush it out and stop attacks in their tracks.

Many cybersecurity tools claim these threats can be thwarted with a combination of artificial intelligence (AI) and automation. But AI is only as good as the model on which it's built, and it can't truly replace humans. Oftentimes, AI and automation lose to human beings because we're unpredictable and not bound by a specific set of rules—which is exactly how hackers operate when they attack. We can't fight unpredictability with a set of rules. Instead, it must be met with human ingenuity.

Persistence isn't a problem we can automate away. Ultimately, cybersecurity is a fight between humans—and sometimes we need to call in the humans to hunt down what automated tools cannot.



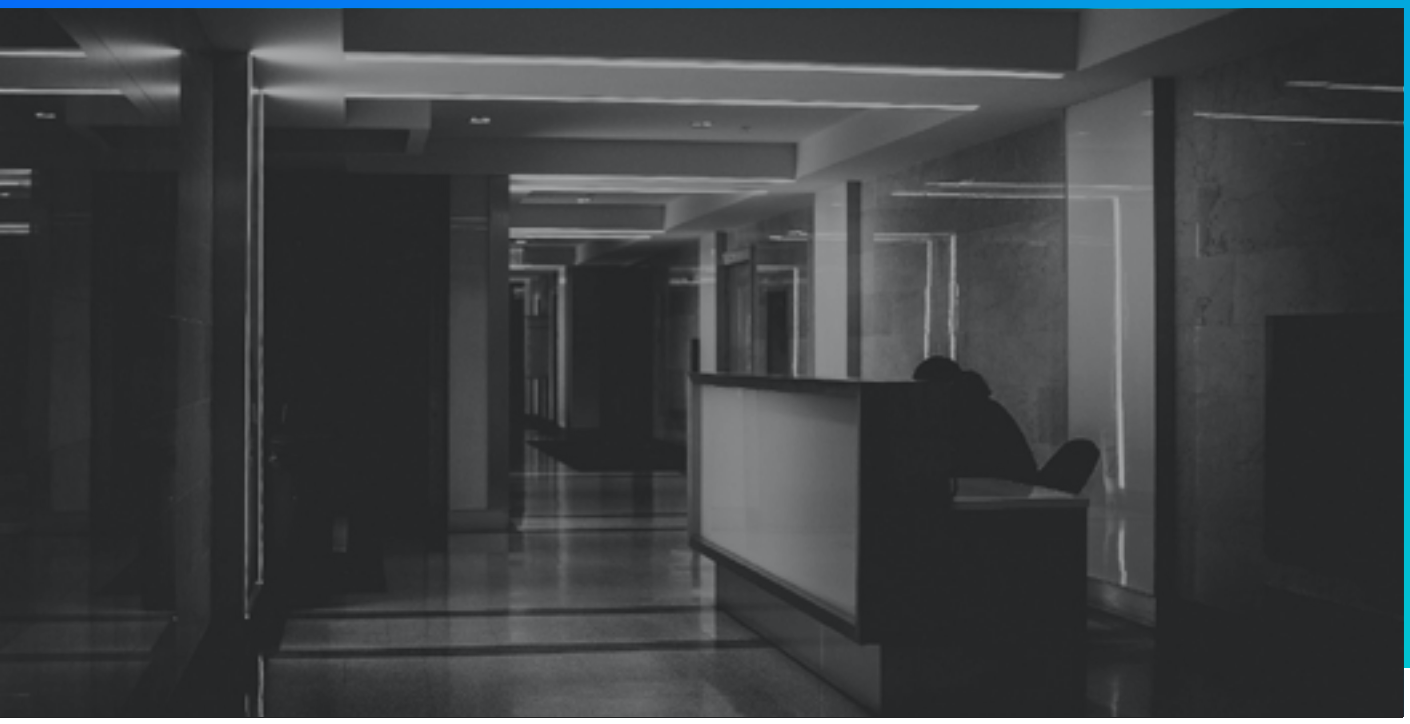
HUMAN THREAT HUNTING

Finding persistence mechanisms requires threat hunting and intelligence—and not the artificial kind.

Threat hunting takes a more offensive approach to security—combining innovative technology with human intelligence to identify attacks that are missed by automated security tools alone. It's more than simply setting off an alarm like many tools do. It's like an alarm with brains. Imagine if a fire alarm had the ability to detect a fire, pinpoint its source and path, alert the building occupants and fire department and forward intelligence to the firehouse prior to their response. That's what threat hunters can do in cybersecurity.

The key to threat hunting is contextual awareness. Some forms of obfuscation or evasion techniques can easily slip past automated solutions, as we commonly see with persistence. A real human being can do the detective work to root out where attackers have established footholds and eliminate their access before they're able to do major damage.

Hunting for persistence is a key differentiator in both the efficiency and efficacy of threat detection and response—which is exactly why Huntress laid down its roots in identifying and eliminating persistent footholds.



OUR FOOTHOLD PHILOSOPHY

Huntress was built on a simple premise—to force hackers to earn every single inch of their access. We've made it our mission to help businesses and IT teams find and eliminate attackers who have silently infiltrated—and are maintaining access to—their environments.

When the game is savvy attackers versus software, attackers win every time. When it's savvy attackers versus Huntress' SOC, the script gets flipped.

Human threat hunters are the backbone of the Huntress platform. Our SOC analysts are trained to look into potential threats, analyze hacker tradecraft, create incident reports and help remediate cyber threats. They understand how attackers operate and put their knowledge and skills to use to defend against new threats and hunt hackers down.

Without the ability to detect and respond to persistent threats, businesses—and the providers that help manage their IT environments—are being attacked without warning and suffering significant damage. It's time to take a more offensive approach to cybersecurity and beat hackers at their own game.

Persistence is a hacker's safety net—Huntress can be yours.

Start your free trial today and let Huntress find and eliminate the persistent threats that are lurking on your Windows and Mac endpoints.

HUNTRESS

huntress.com @HuntressLabs Huntress Huntress