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 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.

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.

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.

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.