2026

Beyond the Honeypot: Increasing Adversarial Cognitive Load Through Interactionless Cyber Deception

Marcus Rozier
Utah State University

Connor Dedic
Utah State University

Ted McDaniel
Utah State University

Hadley Westover
Utah State University

Beyond the Honeypot: Increasing Adversarial Cognitive Load through Interactionless Cyber Deception

Marcus Rozier, Connor Dedic, Ted McDaniel, Hadley Westover
Space Dynamics Laboratory

Abstract

Traditional cyber deception mechanisms, such as honeypots and honeytokens, typically rely on adversary interaction to trigger alerts and provide defensive value. This paper introduces the concept of **interactionless deception**—a defensive paradigm focused on the strategic manipulation of the environment to disrupt the cyber kill chain without requiring direct engagement. By utilizing a greedy top-k selection method to analyze MITRE ATT&CK® data across 178 Advanced Persistent Threat (APT) groups, this research identifies seven high-frequency sub-techniques that represent common attacker behavior. To counter these, the authors developed a series of proof-of-concept tools, including deceptive web proxies, simulated command-and-control (C2) beacons, and endpoint security simulations. These tools are designed to increase the cognitive load on the adversary, inducing psychological effects such as confusion and caution to increase the operational cost of the attack and enhancing the defender's ability to detect and neutralize threats.

Contents

1 Introduction 1
2 Rationale 1
3 Background 1
3.1 What is Deception? 1
3.2 The Deception Mindset 2
3.3 Deception in Attack and Defense 2
3.4 Common Deception Techniques 2
3.4.1 Honey Pots and Honey Nets 2
3.4.2 Credential and Data Decoys 2
3.4.3 Service and Protocol Emulation 3
4 Initial Research 3
4.1 Interactionless Deception 3
4.1.1 Psychological Effects of Interactionless Deception 3
4.2 MITRE ATT&CK Mappings 4
4.2.1 MITRE ATT&CK Analysis Summary 5
5 Results 7
5.1 Deceptive Web Proxy 7
5.2 Simulated Command-and-Control (C2) Beacon 8
5.3 Deceptive Endpoint Security Simulation 8
5.4 Automated Adversary Emulation 9
5.5 POC Demo Tool 9
6 Next Steps 9
Bibliography 11

Figures

Figure 1 Finding Optimal Number of Sub-Techniques to Implement from MITRE ATT&CK 6
Figure 2 Downselecting Sub-Techniques Based on Commonality Amongst Threat Actors 6

Tables

Table 1 MITRE ATT&CK Adversarial Tactics With Current Deception Techniques, Highlighted ... 5
Table 2 MITRE ATT&CK Sub-technique Identifiers and Names 7

1 Introduction

Cyber defenders are constantly trying to find new ways to counter the evolving efforts of adversaries. Popular defense mechanisms like honeypots[1] rely on the concept of deception to lure adversaries in. Most deception techniques require the adversary to interact with them to be effective. This whitepaper explores the novel concept of *interactionless deception*. This deception ideology focuses on creating tools that are easily visible to an attacker in an environment without explicit interaction from them. Research was done to determine which tactics, techniques, and procedures (TTPs) from MITRE ATT&CK® should be targeted for these new tools to emulate to create maximal impact on attacker activities. The addition of these new tools has three distinct goals for the adversary: slow them down to re-think their attack path, speed them up to cause mistakes, or make them question their presence on the system entirely.

2 Rationale

The inspiration to innovate on current industry deception techniques was inspired by a paper written by Jared Chandler of Tufts University and Adam Wick of Fastly [2]. They hypothesized that a "self-attack" program emulating an attacker on a system would successfully deter, distract, and disrupt a real adversary.

This team did a semi-formal study which demonstrated that in environments with this deception technique employed, emulated attackers were less successful and more cautious. This was an effective deception technique that didn't require direct adversary interaction for it to be successful. This whitepaper sought to expand on this concept and see what other interactionless deception tools could realistically fool an adversary.

3 Background

3.1 What is Deception?

Deception[3] in the context of this paper refers to the deliberate placement of fabricated assets—such as honeypots, honey-networks[4], bogus credentials, and decoy services[5]—to mislead adversaries and influence their behavior. By presenting attractive but non-operational resources, defenders can lure attackers away from critical systems, gain early visibility into intrusion attempts, collect intelligence on TTPs, and ultimately reduce the risk of compromise. In essence, deception creates controlled, observable "leads" that turn an attacker's curiosity into a defensive advantage.

Building on that foundation, the primary purpose of deception is three-fold:

1. Early Detection
2. Controlled Engagement
3. Actionable Intelligence.

By inserting believable but isolated lures throughout the network, defenders gain an indicator of compromise the moment an adversary interacts with a decoy—often well before any legitimate asset is touched. Once engaged, the attacker is confined to a sandboxed[6] environment where their movements can be safely monitored, allowing security teams to study TTPs without risking production systems. The data harvested from these interactions fuels threat-intel feeds, refines detection signatures, and informs incident-response playbooks, while the very presence of deceptive elements forces threat actors to waste time, resources, and confidence, thereby reducing the overall likelihood of a successful breach.

3.2 The Deception Mindset

The most important aspect of implementing deception in a network is developing a deception mindset. A deception mindset treats the network not as a static fortress but as a dynamic battlefield where every asset can serve as either a shield or a lure. Defenders start by thinking like the adversary—identifying the data, services, and pathways that would be most enticing, then deliberately reproducing them in a controlled and isolated fashion. This requires a proactive stance: instead of waiting for signatures or alerts, the security team engineers "sweet spots" that draw attention. They then continuously map how attackers might pivot and adjust the decoys to stay relevant as tactics evolve. In this view, deception is not an after-thought add-on. It is baked into architecture, change-management, and risk-assessment processes from day one.

Equally important is the feedback-driven culture that underpins the mindset. Every interaction with a decoy is logged and analyzed to extract TTPs, which then feed back into detection rules, threat-intel feeds, and incident-response playbooks. Success is measured not by the number of fake assets deployed, but by metrics such as time-to-detect[7], containment duration[8], and the enrichment of threat intelligence. Teams adopt an iterative loop: deploy, observe, refine, and redeploy—much like an A/B test for security—ensuring that deception remains a living, adaptive layer that continuously raises the cost and uncertainty for any would-be intruder.

3.3 Deception in Attack and Defense

Attackers employ deception to hide their presence, mask intent, and increase the odds of a successful breach. Common tactics include impersonating legitimate users or services with forged credentials, deploying living-off-the-land[9] tools that blend into normal system processes, and using tunnelled or steganographic channels to exfiltrate[10] data unnoticed. By planting false flags—such as fake error messages, bogus configuration files, or decoy data—they can mislead blue-team investigations, steer forensic analysts toward irrelevant artifacts, and buy time to establish persistent footholds. In essence, the attacker's deception agenda is to create uncertainty for the defender, making it harder to distinguish benign activity from malicious behavior.

Defenders turn deception into a proactive weapon. By strategically deploying honeypots, honey-networks, bogus credentials, and deceptive services, they generate high-confidence alerts the moment an adversary interacts with a lure. This early warning system forces attackers into a controlled engagement where every command, lateral move, and data request can be captured and analyzed without endangering production assets. The intelligence harvested informs defensive strategies while the decoys incur more time spent probing, more resources expended on false leads, and a higher risk of exposure. In this cat-and-mouse game, only the defender can dictate the terms of the interaction, turning the attacker's own preferred tactic against them.

3.4 Common Deception Techniques

3.4.1 Honey Pots and Honey Nets

Honeypots are isolated systems that mimic real services (web servers, databases, IoT[11] devices, etc.) but contain no production data. When an adversary connects, every interaction is logged and the environment can be safely sandboxed. A honey-network interconnects several honeypots to emulate an entire subnet, complete with believable internal traffic, routing tables, and DNS records. This gives defenders a richer view of lateral-movement tactics while keeping the attacker far from critical assets.

3.4.2 Credential and Data Decoys

Fake usernames, passwords, API[12] keys, and tokens are deliberately seeded in configuration files, code repositories, or secrets dumps. When an attacker harvests or attempts to use these "golden"

credentials, an alert fires. Similarly, decoy documents (e.g., fabricated financial spreadsheets, design schematics, or PII files) are watermarked or instrumented with tracking beacons; opening or exfiltrating them reveals the threat actor's intent and provides forensic breadcrumbs without exposing genuine information.

3.4.3 Service and Protocol Emulation

Deception can also masquerade as legitimate network services—SSH, RDP, SMB, Kubernetes APIs, or cloud-provider endpoints—by running low-interaction emulators that respond just enough to be convincing. Advanced emulators replay realistic banner strings, latency patterns, and even scripted error conditions, fooling automated scanners and manual recon. Protocol-level tricks such as fake TLS certificates, "ghost" DNS entries, or bogus cloud metadata endpoints lure attackers into requesting authentication tokens or exposing their tooling, giving defenders a chance to capture and analyze the payloads before they ever reach real infrastructure.

4 Initial Research

4.1 Interactionless Deception

The current state of cyber deception prioritizes creating a separate plane and directing the attacker to it. The idea of this interactive deception is to focus an attacker into a real system that isn't a risk to the network and is designed to alert the security team of the attackers's presence. However, focusing solely on shifting an attacker's focus away from operational systems has it's limits. When an attacker enters a system, they search for any information about the system that they can. They search documents, settings, and watch network traffic for anything they can use to exploit systems. All of those components are things that can be used to deceive attackers. There is no reason to limit deception to specific systems or canaries. The whole system can be used to the defender's advantage. People often say that attackers have the advantage with time, but defenders have the advantage of the terrain. The basis of interactionless deception is simple: create an environment that is dynamically changing or even appears to be already compromised to interrupt the normal flow of the cyber kill chain. The environment itself serves as the deceptive defense without the need for interaction with specific dummy systems for success.

While this is a new concept within cyberspace, interactionless deception has been demonstrated in military tactics for years. In the past, battles were won by the force with the tightest formations and best battle plans. Over the last 100 years we have seen a different trend. Guerrilla warfare has proven to be almost impossible to win against in prolonged campaigns.[13] Traditional cyber defenses prioritize utilizing structures (like network segments, bastions and firewalls) as primary defenses. But how would attackers treat an environment when any system could be a trap? How would they react when any document may ping their location, when any command may return a less then truthful output or when they see another threat actor attacking the system they are on will they know if it is real? The primary paradigm of defending a cyber system and utilizing guerilla warfare is the same, win not in a decisive victory but by increasing the cost of the attack to the point that it is no longer economical to continue. Even if a deception mechanism doesn't directly correlate to alerts, it will affect how long it takes them to move through the system unnoticed. In turn, adversarial operations take more time, increase exposure, and cost more money.

4.1.1 Psychological Effects of Interactionless Deception

Many psychological effects can be triggered to an attacker's mind when interactionless deception is present in the systems being penetrated. These effects include:

- Confusion
- Caution

- Fear
- Frustration
- Stress

Each of these psychological effects can decrease the attacker's effectiveness and increase the possibility of failure in the attacker meeting their goals. The attacker may experience confusion in not being able to get a clear mapping of the system. This confusion will make the attacker unsure on which systems contain exploitable vulnerabilities for accomplishing their goals. By not having a clear knowledge of the system, an attacker will be hesitant to act in the fear of messing up and causing detection. This may even cause the attacker to give up on the target and instead switch to a new target that will provide a simpler path to success.

If an attacker suspects or see evidence that deception or monitoring is being used on a system, it will greatly increase their level of caution in discovering and exploiting that system. This caution slows down an attacker in their work and may cause hesitancy in executing steps that would lead to success.

In a 2024 study[14] researchers prepared two similar environments for penetration testers to attack. The difference in the environments was one contained cyber deception measures and the other did not. The results of this study found that penetration testers working in the deceptive environment were 170% slower at completing the reconnaissance phase and there was an 80% decrease in penetration testers who were able to complete the objective of compromising the system. When interviewing the penetration testers after the exercise it was found that they thought the deceptive environment was much more difficult and more stressful because they felt they had to be sneaky. This study demonstrates the effectiveness that can come from attackers experiencing psychological effects of deception while trying to penetrate a system.

4.2 MITRE ATT&CK Mappings

The MITRE ATT&CK®[15] framework is a cybersecurity knowledge base that contains 14 unique adversarial tactics:

During any adversarial cyber engagement, multiple exploitation techniques under each of these tactics will be attempted. The question for defenders becomes this: How can these different techniques be slowed / stopped completely to keep the operating environment safe? Under the 14 tactics of the MITRE ATT&CK Framework discussed above, there are a total of 250 attack techniques that exist. To visualize the coverage of defensive deception on these techniques, tables were created which placed an attack next to a deception (if applicable) to see which categories had stronger defensive coverage. While there are some areas where deception can really thrive on the defensive, it became apparent that current deception techniques were either not the best fit or flat out incapable of having the desired stopping/slowing effect in all of the 14 tactics. Table 1 highlights in green which techniques would generate a positive defensive effect within the context of current deception techniques.

Table 1: MITRE ATT&CK Adversarial Tactics With Current Deception Techniques, Highlighted

Reconnaissance	Resource Development
Initial Access	Execution
Persistence	Privilege Escalation
Defense Evasion	Credential Access
Discovery	Lateral Movement
Collection	Command and Control
Exfiltration	Impact

NOTE: This table was created before the April 2026 update to the MITRE ATT&CK framework.

Reconnaissance: Reconnaissance consists of techniques that involve adversaries actively or passively gathering information that can be used to support targeting[15]. Honeypots, honeytokens, and canary documents are examples of current deception practices that can taint an attackers information from reconnaissance.

Resource Development: Resource Development consists of techniques that involve adversaries creating, purchasing, or compromising/stealing resources that can be used to support targeting[15]. Defenders can release fake architectural documents to deceive attackers into which resources they need to be able to penetrate into the system.

Initial Access: Initial Access consists of techniques that use various entry vectors to gain their initial foothold within a network[15]. Honeytokens and canary documents can provide an attacker with credentials to gain initial access, but into an isolated, controlled environment.

Discovery: Discovery consists of techniques an adversary may use to gain knowledge about the system and internal network[15]. Honeypots, decoy hosts, and honey files are deception practices that prevent the attacker from accurately discovering the internal network and systems.

There are still 10 additional tactics where deception as it stands today is insufficient which still leaves a significant portion of the 250 possible attack avenues. This analysis is helpful for demonstrating where deception fits into the framework at a high level, but more value can be drawn from a deeper analysis of the most common techniques that are being employed by adversaries today.

4.2.1 MITRE ATT&CK Analysis Summary

Using the MITRE ATT&CK mapping, research was done to discover which techniques are used most commonly by attackers. To do this, a metric was built using a greedy top-k method that searches for the smallest set of sub-techniques that are utilized across the largest number of Advanced Persistent Threat (APT) groups. Instead of focusing only on which techniques are used most often, the goal was to find which techniques best represent common attacker behavior overall. The final set size was chosen when adding more sub-techniques improved the results by less than 1%. The resulting group represents the techniques most commonly associated with real-world attacks and most likely to appear together. The algorithm starts by finding the single sub-technique used by the most APT groups. It then repeatedly adds the next sub-technique that covers the largest number of additional groups not already represented. This continues until adding more sub-techniques increases the total coverage by less than 1%.

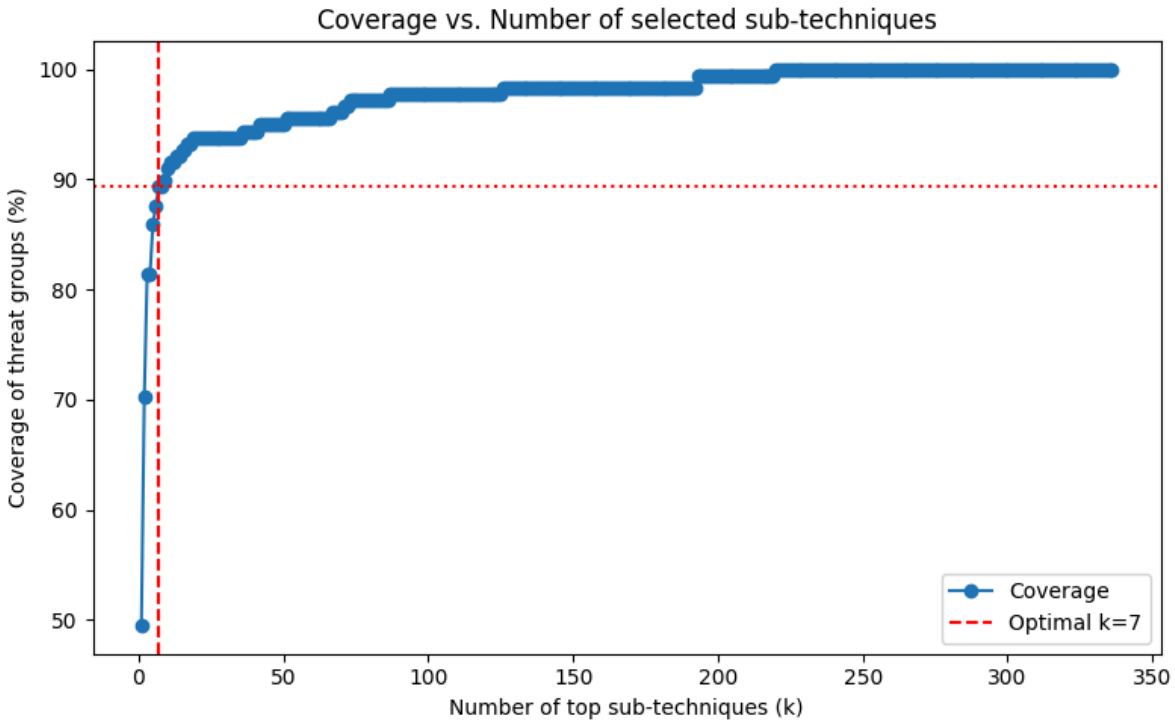

Figure 1: Finding Optimal Number of Sub-Techniques to Implement from MITRE ATT&CK

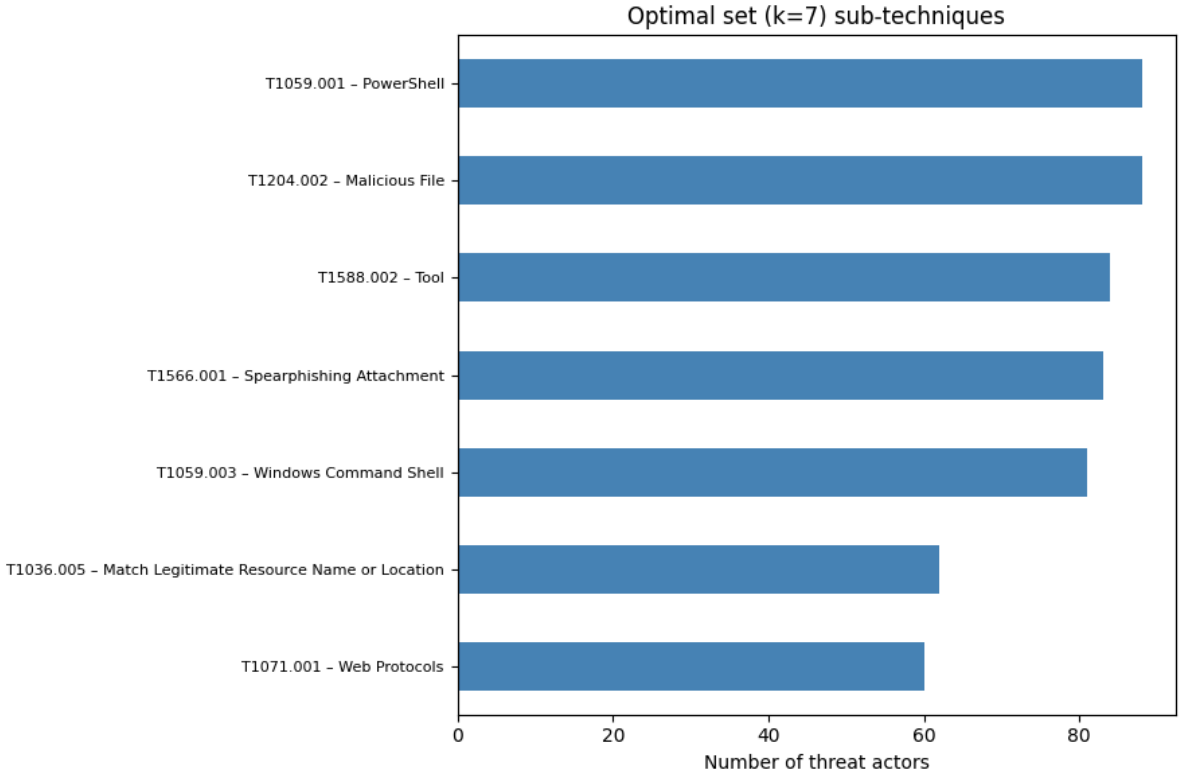

Figure 2: Downselecting Sub-Techniques Based on Commonality Amongst Threat Actors

The graph in Figure 2 shows the total frequency of the sub-techniques we looked at. The set was similar to frequency, but was found to have different techniques. A frequency group would show most common techniques and the greedy top-k shows the combination of that best represents attacker behavior. This data used 178 APT groups, with a total of 336 distinct techniques. We found

that the group of [‘T1204.002’, ‘T1059.001’, ‘T1588.002’, ‘T1566.001’, ‘T1059.003’, ‘T1036.005’, ‘T1071.001’] had 89.3% over all the threat groups. Table 2 correlates these sub-technique identifiers with their names.

Table 2: MITRE ATT&CK Sub-technique Identifiers and Names

Sub-technique ID	Sub-technique Name
T1204.002	User Execution: Malicious File
T1059.001	Command and Scripting Interpreter: PowerShell
T1588.002	Obtain Capabilities: Tool
T1566.001	Phishing: Spearphishing Attachment
T1059.003	Command and Scripting Interpreter: Windows Command Shell
T1036.005	Masquerading: Match Legitimate Resource Name or Location
T1071.001	Application Layer Protocol: Web Protocols

With these critical techniques identified, the team was able to focus on creating interactionless deception implementations can could provide defensive advantages against these techniques.

5 Results

To empirically validate the viability of the proposed theoretical framework, this section presents a series of proof-of-concept (PoC) implementations. By translating abstract deception strategies into operational tools, this work demonstrates how strategic misinformation can be systematically applied to neutralize attacker advantages and erode their confidence in the target environment. While these implementations represent an initial exploration rather than an exhaustive catalog of all interaction-less deception techniques, they effectively bridge the gap between theoretical conceptualization and practical application. The following tools serve as a foundational catalyst for the development of more sophisticated, scalable, and autonomous deceptive architectures in modern cybersecurity.

5.1 Deceptive Web Proxy

Upon initial compromise, adversaries typically engage in reconnaissance to fingerprint the target environment and map the attack surface. This process involves both manual probing and automated scanning to identify operating systems, services, and network topologies. The proposed proxy intercepts this fingerprinting traffic in real-time, dynamically modifying responses to inject misleading metadata. By altering operating system identifiers and simulating the presence of non-existent services, the tool creates a deceptive architectural landscape. This erosion of data integrity forces adversaries to spend significantly more time validating their findings, thereby increasing the cognitive load on the attacker and slowing the progression of the kill chain. Furthermore, all intercepted interactions are comprehensively logged, providing defenders with high-fidelity telemetry regarding the adversary’s intent and methodology.

The successful implementation of this tool will impact three of the top seven MITRE sub-techniques:

- Web Protocols - As each scan and environmental interaction reveals different protocols and services, adversaries will find it difficult to isolate the legitimate system configuration.
- Match Legitimate Resource Name or Location - As each scan and environmental interaction reveals different protocols and services, adversaries will find it difficult to isolate the legitimate system configuration.
- Tool - As automated / manual tools attempt to map the environment, the variability of protocols and services will hinder the isolation of the legitimate system configuration.

5.2 Simulated Command-and-Control (C2) Beacon

Upon establishing a foothold, adversaries typically perform process enumeration to ascertain system functionality and identify candidate processes for code injection or the establishment of persistence. A critical component of this persistence is the Command-and-Control (C2) beacon—a persistent process that enables remote orchestration and data exfiltration. This tool implements a deceptive C2 beacon that mimics these characteristics without facilitating any actual unauthorized data transfer. To maintain authenticity, the tool generates synthetic network traffic that replicates the behavioral patterns and heartbeat intervals of established C2 frameworks. The presence of this simulated beacon is designed to induce a perception of prior compromise, leading the adversary to conclude that the system is already under the control of a competing threat actor. This creates a strategic dilemma for the attacker: they may attempt to rapidly deploy their own C2 infrastructure to maintain dominance—thereby increasing the likelihood of operational errors—or they may attempt to terminate the existing beacon to clear the environment.

The successful implementation of this tool will impact three of the top seven MITRE sub-techniques:

- PowerShell - The appearance of an active compromise forces adversaries to modify their traditional use of PowerShell in the environment. By introducing this uncertainty, we compel the attacker to either slow down (increasing the window for detection), rush their objectives (increasing the probability of detectable mistakes), or a complete withdrawal from the environment.
- Windows Command Shell - The appearance of an active compromise forces adversaries to modify their traditional use of CMD in the environment. By introducing this uncertainty, we compel the attacker to either slow down (increasing the window for detection), rush their objectives (increasing the probability of detectable mistakes), or a complete withdrawal from the environment.
- Tool - If automated/manual tooling indicates that an environment appears to be actively compromised, adversaries will be forced to adjust the aggressiveness of their actions on objectives, necessitating more careful planning and slowing their movement

5.3 Deceptive Endpoint Security Simulation

Endpoint security software is fundamental to the detection and remediation of malicious binaries within a host environment. To counter these defenses, modern malware frequently employs anti-analysis and evasion mechanisms designed to detect the presence of antivirus (AV) engines or sandbox environments; if such software is identified, the malware will typically inhibit its own execution to avoid detection. This tool leverages this adversarial behavior by simulating the operational signatures of various security suites, thereby inducing a state of dormancy in the malicious payload. By decoupling the act of detection from the act of remediation, the tool creates a critical window of opportunity for security analysts to isolate and neutralize the threat before the adversary can successfully execute their payload or escalate privileges.

The successful implementation of this tool will impact three of the top seven MITRE sub-techniques:

- Malicious File - Due to the presence of various antivirus and sandbox detection mechanisms, the file's malicious functionality remains dormant to evade detection and analysis.
- Spearphishing Attachment - Due to the presence of various antivirus and sandbox detection mechanisms, the file's malicious functionality remains dormant to evade detection and analysis.
- Tool - Automated / manual tools would detect the presence of various antivirus and sandbox signatures, complicating the process of ascertaining which defenses are actually active in the environment.

5.4 Automated Adversary Emulation

Following the initial breach, an adversary typically initiates a reconnaissance phase to map the host environment and identify high-value targets. This tool disrupts this process by synthesizing adversarial behavior through the automated execution of reconnaissance command sets. To avoid the predictability of standard scripts and successfully mimic human interaction, the tool employs stochastic intervals and sequenced command patterns, effectively replicating the operational cadence of a manual operator. The presence of these concurrent activities is designed to create the illusion of a competing threat actor already operating within the system. This perception of a multi-adversary environment induces cognitive dissonance and operational hesitation in the attacker; the sudden uncertainty regarding the security and exclusivity of their foothold forces them to deviate from their optimized attack path. Consequently, the adversary is compelled to adopt a more cautious, deliberative approach—slowing their progression and increasing the likelihood that their movements will be detected by defensive monitoring systems.

The successful implementation of this tool will impact three of the top seven MITRE sub-techniques:

- PowerShell - The appearance of an active compromise forces adversaries to modify their traditional use of PowerShell in the environment. By introducing this uncertainty, we compel the attacker to either slow down—increasing the window for detection—or rush their objectives, increasing the probability of detectable mistakes or a complete withdrawal from the environment.
- Windows Command Shell - The appearance of an active compromise forces adversaries to modify their traditional use of CMD in the environment. By introducing this uncertainty, we compel the attacker to either slow down—increasing the window for detection—or rush their objectives, increasing the probability of detectable mistakes or a complete withdrawal from the environment.
- Tool - If automated / manual tooling indicates that an environment appears to be actively compromised, adversaries will be forced to adjust the aggressiveness of their actions on objectives, necessitating more careful planning and slowing their movement

5.5 POC Demo Tool

To evaluate the efficacy of these countermeasures, the aforementioned proof-of-concept (PoC) implementations were synthesized into a unified demonstration framework. This environment allows for a comparative analysis of adversarial progression, contrasting the trajectory of an attack in a baseline environment against one where deceptive layers are active.

While these simulations do not encapsulate the full breadth of all potential interaction-less deception strategies, they provide a tangible illustration of the strategic advantages inherent in deceptive cyber defense, demonstrating the potential to disrupt adversarial momentum and significantly enhance detection capabilities. Additionally, the collective efficacy of these integrated proofs-of-concept offer the potential to mitigate the seven primary attack sub-techniques within a deployed environment.

6 Next Steps

The findings presented in this paper serve as a foundational step toward a more robust understanding of interactionless deception. To build upon this baseline, future research will be directed toward three critical objectives.

1. A comparative analysis will be conducted to quantify the effectiveness of interactionless deception against traditional deception strategies.
   - By utilizing a controlled, simulated environment, this study will identify the specific scenarios where interactionless deception excels independently and where traditional methods offer superior defense.

2. The research will explore the implementation of AI-driven orchestration to enable dynamic environment shifts.
    - By autonomously altering system attributes in real-time, this integration will actively distort an adversary's perception of the target environment, thereby increasing the cognitive load on the attacker.
3. The current Proof of Concepts (POCs) will be evolved from basic functional models into high-fidelity implementations, focusing on scalability and seamless integration into existing enterprise security architectures.

Bibliography

[1] Fortinet, “What Are Honeypots (Computing)?.” [Online]. Available: https://www.fortinet.com/resources/cyberglossary/what-is-honeypot

[2] J. Chandler and A. Wick, “Deceptive Self-Attack for Cyber-Defense.” [Online]. Available: https://scholarspace.manoa.hawaii.edu/server/api/core/bitstreams/eaf37340-6c3e-48c3-974b-51f41a811ccb/content

[3] Meriam and Webster, “deception.” [Online]. Available: https://www.merriam-webster.com/dictionary/deception

[4] GeeksForGeeks, “Honeypot vs Honeynet.” [Online]. Available: https://www.geeksforgeeks.org/ethical-hacking/honeypot-vs-honeynet/

[5] Fortinet, “What Is Canary In Cybersecurity?.” [Online]. Available: https://www.fortinet.com/resources/cyberglossary/what-is-canary-in-cybersecurity

[6] Fortinet, “What Is Sandboxing?.” [Online]. Available: https://www.fortinet.com/resources/cyberglossary/what-is-sandboxing

[7] SentinelOne, “What Is MTTD (Mean Time to Detect)? A Detailed Explanation.” [Online]. Available: https://www.sentinelone.com/blog/mttd-mean-time-to-detect-detailed-explanation/

[8] AWS, “Containment.” [Online]. Available: https://docs.aws.amazon.com/security-ir/latest/userguide/containment.html

[9] B. Lenaerts-Bergmans, “What Are Living off the Land (LOTL) Attacks?.” [Online]. Available: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/living-off-the-land-attack/

[10] Fortinet, “What Is Data Exfiltration?.” [Online]. Available: https://www.fortinet.com/resources/cyberglossary/data-exfiltration

[11] IBM, “What is the Internet of Things (IoT)?.” [Online]. Available: https://www.ibm.com/think/topics/internet-of-things

[12] I. Cornea, “What Are APIs? A Beginner's Guide (with examples).” [Online]. Available: https://dev.to/icornea/what-are-apis-a-beginners-guide-with-examples-4ok8

[13] Andrew Mack, “Why Big Nations Lose Small Wars: The Politics of Asymmetric Conflict.”

[14] B. Bowman, T. Thomas, and J. Kaabi, “Cyber Deceptive Countermeasures’ Effects on Human-Simulated Attacks.” [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10778664

[15] MITRE, “MITRE ATT&CK®.” [Online]. Available: https://attack.mitre.org/