digiDations

Rethinking Security Validation in Modern Enterprise Environments

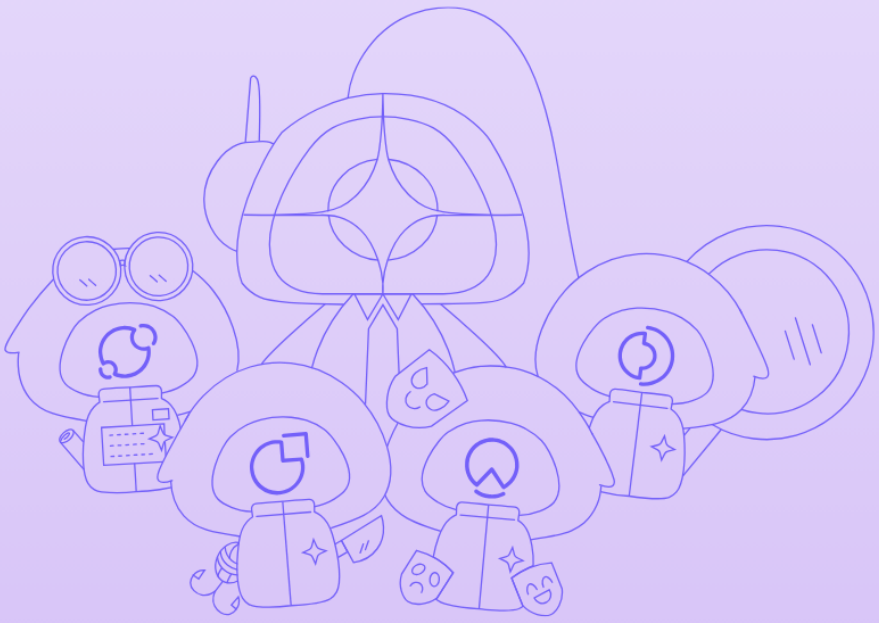

Table of Contents

Introduction

Modern enterprise environments rely on a wide range of security technologies, including network security controls, endpoint protection platforms, and security monitoring systems. These tools are designed to detect, prevent, and respond to cyber threats across different stages of the attack lifecycle. While organizations invest heavily in deploying such defenses, an important question remains difficult to answer with confidence:

Are these defenses capable of stopping real-world attacks when they occur?

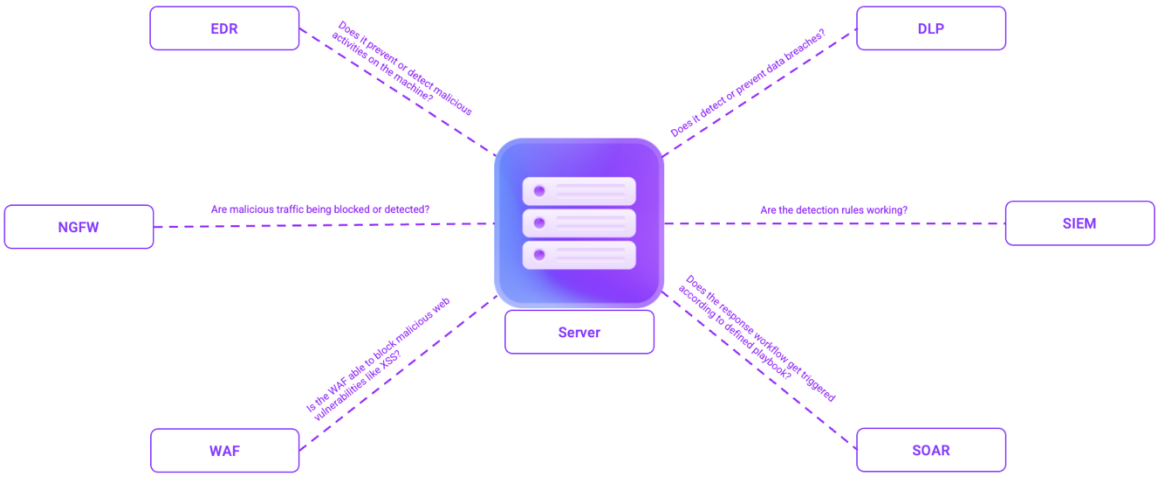

Figure 1 - Types of Security Control on a Server

Traditional security assessments such as vulnerability scanning, penetration testing, and red teaming provide valuable insights into potential weaknesses. However, these approaches often represent point-in-time evaluations and may not provide continuous visibility into the effectiveness of deployed security controls. As enterprise environments evolve and threat actors adopt increasingly sophisticated tactics, organizations require more systematic ways to validate whether their defenses remain effective against modern attack techniques.

Breach and Attack Simulation (BAS) technologies have emerged as one approach to addressing this challenge. By simulating attacker behaviors and techniques within an enterprise environment, BAS solutions aim to help organizations evaluate whether security controls can detect or block specific attack activities. In principle, this approach enables defenders to validate their security posture in a safe and repeatable manner.

However, implementing realistic and practical attack simulation presents several technical and operational challenges. Traditional BAS architectures often rely on vulnerable target hosts or

simplified attack demonstrations, which can introduce security risks, operational overhead, and limitations in accurately simulating modern attack chains.

This paper examines key challenges associated with traditional BAS approaches and discusses technical methods that can help address these limitations. In particular, it explores how packet replay techniques and sandbox-based execution of real malware samples can support safer and more comprehensive security validation, enabling organizations to better assess the effectiveness of their defenses across multiple stages of the attack lifecycle.

Challenges in Traditional Breach and Attack Simulation

The Target Host Dilemma

Many traditional attack simulation approaches assume that two components are required: an attacker system and a target host. In this model, attack scripts and payloads are placed on the attacker system, while the target host is typically a virtual machine or container intentionally configured with vulnerable operating systems or applications. The attacker launches the exploit against the target host, and if the exploit succeeds, the attack simulation is considered complete.

While conceptually straightforward, this approach introduces several significant challenges:

1. Security risks introduced by vulnerable targets

Target hosts used for simulation are intentionally vulnerable and unpatched. Introducing such systems into an enterprise environment can create additional security risks, which are often unacceptable in production networks. Even if the testing environment is intended to be isolated, strict isolation cannot always be guaranteed.

If a malicious actor already exists within the network, even a short-lived vulnerable system—running for only a few minutes—may provide an opportunity for compromise. An attacker could exploit the system and potentially use it as a foothold for lateral movement or other malicious activities.

2. High operational and maintenance overhead

New vulnerabilities are continuously discovered, requiring target hosts to be regularly rebuilt and updated. Maintaining these environments introduces significant operational overhead and ongoing maintenance effort.

Although vendors may provide prebuilt images or update packages, managing these assets at scale remains complex and resource intensive. In highly restricted environments such as air-gapped networks, distributing and updating these images can be particularly challenging.

3. Limited ability to simulate modern attack chains

Modern cyberattacks, especially those associated with Advanced Persistent Threats (APTs), rarely rely on a single vulnerability exploit. Instead, attackers employ a sequence of tactics, techniques, and procedures (TTPs) across multiple stages of the attack lifecycle.

The traditional attacker–target model focuses primarily on vulnerability exploitation and therefore struggles to accurately simulate the broader sequence of activities that occur throughout a real attack campaign.

4. Increased deployment complexity

To simulate attack paths across multiple network segments or security zones, both attacker and target hosts often need to be deployed within each zone. This effectively doubles the required infrastructure and significantly increases deployment complexity, operational cost, and management overhead.

The Real Malware Execution Dilemma

Accurately validating security defenses against modern threats ideally requires executing real attack techniques, including the use of genuine malware samples. In real-world intrusions, malware performs a variety of complex behaviors such as process injection, privilege escalation, command-and-control communication, and destructive actions including data encryption or system disruption. Testing security controls against these authentic behaviors would provide a realistic assessment of whether defensive technologies can detect or prevent malicious activity.

However, executing real malware within enterprise environments introduces significant security risks. Malware samples are designed to perform harmful actions, and if security controls fail to detect or contain them during testing, the consequences could include system damage, operational disruption, or unintended propagation across the network. Even when testing environments are intended to be isolated, strict containment cannot always be guaranteed, particularly in complex production infrastructures.

Because of these risks, many Breach and Attack Simulation (BAS) platforms avoid executing real malware samples. Instead, they rely on demonstration programs or simplified payloads designed to imitate certain malicious behaviors. For example, a simulated ransomware program may encrypt files in a controlled manner in order to trigger detection mechanisms.

While these approaches reduce operational risk, they introduce important limitations. Demonstration programs cannot fully replicate the structure, execution flow, and behavioral patterns of genuine malware. As a result, security products may respond differently to simulated payloads than they would to real threats.

This limitation becomes particularly significant during the **Execution** and **Impact** stages of the attack lifecycle. These phases often involve complex behaviors that are difficult to accurately reproduce without executing real malicious code. Consequently, many traditional BAS

solutions struggle to simulate the full attack kill chain and may not reliably answer a key question for enterprise defenders:

Would existing security controls detect or stop a real attack at the moment it occurs?

The challenge of safely executing real malware therefore represents a fundamental barrier to achieving realistic and comprehensive security validation.

Technical Approaches for Practical Security Validation

Packet Replay for Network Attack Simulation

Packet replay technology has traditionally been used in network forensics, protocol analysis, and performance testing. The same underlying principle can also be applied to attack simulation scenarios in order to evaluate the effectiveness of network security controls.

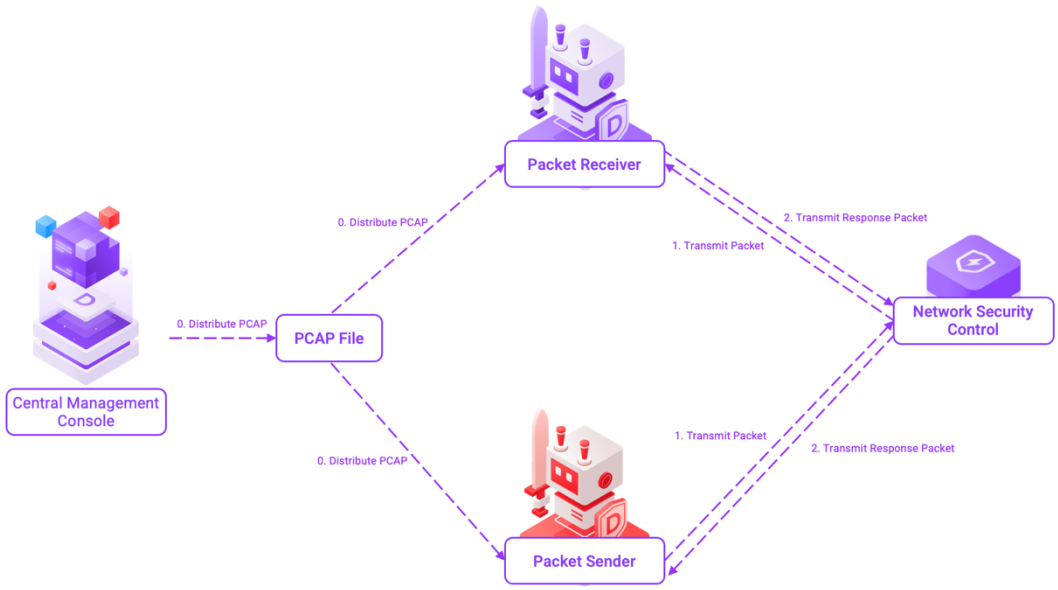

Figure 2 - Packet Replay for Network Attack Simulation

A packet replay–based validation approach typically involves three components: a packet sender, a packet receiver, and a central management console. The console distributes pre-recorded network traffic in the form of PCAP files that contain captured attack sequences, malicious communication patterns, or other threat-related network behaviors.

The sender and receiver extract the relevant packet data from the PCAP files and reproduce the network traffic according to the original packet sequence. The sender transmits packets toward the receiver while the receiver provides the expected responses defined in the captured traffic sequence. By replaying both sides of the communication flow, the system can reconstruct the behavior of a real attack session without requiring a vulnerable target host.

During the replay process, the sender and receiver continuously verify whether the observed responses match the expected packet sequence.

- If the received responses match the expected behavior defined in the PCAP data, the replay continues to the next stage of the attack sequence.
- If the responses differ from the expected pattern, this indicates that a security control has intercepted, modified, or blocked the traffic, and the simulated attack session is terminated.

To ensure accurate detection by network security devices, the packet sequence, structure, and timing must closely follow the original captured traffic. Simply sending packets without preserving these characteristics may prevent security controls from correctly identifying the threat indicators embedded in the traffic.

By simulating attack behavior through packet replay, organizations can validate the effectiveness of network-based security controls without deploying vulnerable target hosts. This approach reduces operational complexity and eliminates the security risks associated with maintaining intentionally vulnerable systems in production environments.

Sandbox-Based Execution of Real Malware Samples

Sandbox technology has long been used in malware analysis workflows. In a typical security environment, suspicious or unknown files are submitted to a sandbox through an API, where the sample is executed in an isolated environment. The sandbox observes the behavior of the program and generates an analysis report describing its activities and potential malicious characteristics.

From a technical perspective, sandbox technology can also be adapted to support attack simulation and security validation.

In this model, malware samples from a curated attack library are executed inside a controlled sandbox environment designed specifically for validation purposes. The sandbox provides strong isolation and containment capabilities to ensure that the executed malware cannot escape the environment or affect external systems.

However, enabling reliable execution of malware samples requires more than simply running the file inside an isolated environment. Many malware programs rely on specific environmental conditions, configuration parameters, or supporting files to trigger their intended behavior. As a result, BAS vendors must perform detailed malware analysis and reverse engineering in order to understand how each sample operates.

Through this preparation process, the required environmental conditions can be reproduced within the sandbox so that the malware executes its intended behavior when triggered by the validation system.

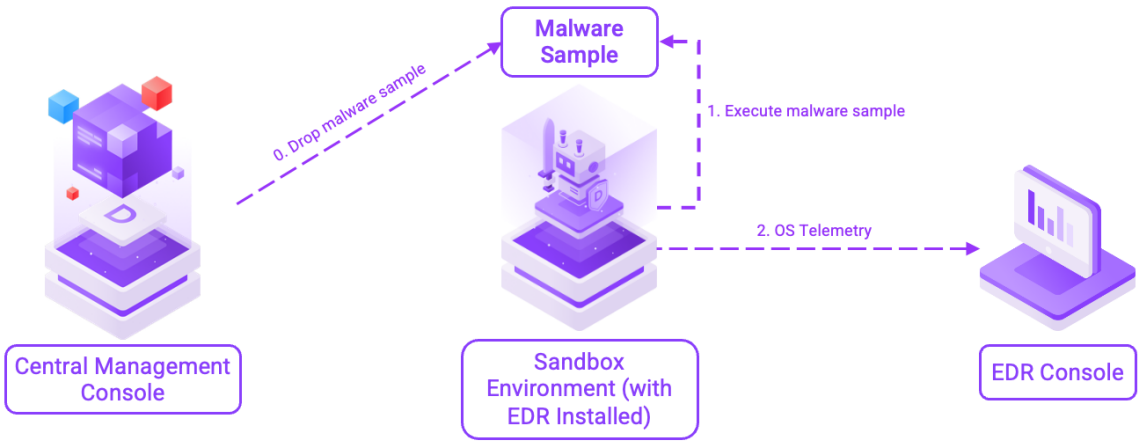

Figure 3 - Executing Malware on Isolated Environment

In addition, the sandbox environment can include the same endpoint security products used within the enterprise environment, such as antivirus (AV) or endpoint detection and response (EDR) solutions. This allows organizations to directly observe whether their deployed security controls detect or block the malware at the moment of execution.

By enabling the safe execution of real malware samples, sandbox-based simulation makes it possible to test security controls against authentic malicious behavior while maintaining strong containment safeguards. This capability helps address the limitations of traditional attack simulation approaches and enables more comprehensive validation across the **Execution** and **Impact** stages of the attack lifecycle.

Implications for Enterprise Security Validation

The technical approaches described in the previous section address key limitations associated with traditional Breach and Attack Simulation (BAS) architectures. By eliminating the need for vulnerable target hosts and enabling the safe execution of real malware samples, organizations can perform security validation in a manner that is both safer and more representative of real-world attack conditions.

Safer Validation in Operational Environments

One of the primary barriers to security validation in enterprise environments is the operational risk introduced by testing activities. Traditional approaches that rely on vulnerable systems or uncontrolled execution of malicious code can introduce security and operational concerns, particularly in production environments.

Packet replay–based simulation reduces this risk by removing the requirement for intentionally vulnerable target hosts. Instead of deploying exploitable systems within the network, attack behaviors can be reproduced through controlled network traffic sequences derived from real attack patterns. This allows organizations to validate the effectiveness of network security controls—such as firewalls, intrusion detection systems, and network detection and response solutions—without introducing additional attack surfaces.

Similarly, sandbox-based execution environments enable the controlled execution of real malware samples while maintaining strict isolation from the production environment. By confining malicious behavior within a sandbox, organizations can observe how endpoint security products respond to authentic threats without exposing operational systems to risk.

Together, these techniques make it possible to conduct meaningful security validation while maintaining the operational safety required in enterprise environments.

Enabling Full Attack Lifecycle Validation

Modern cyberattacks typically involve a sequence of activities that span multiple stages of the attack lifecycle. These stages may include initial access, execution, persistence, lateral movement, command-and-control communication, and ultimately impact or data exfiltration.

Validating defenses against isolated attack techniques provides limited insight into an organization's overall defensive posture. Instead, effective security validation should consider how defensive technologies perform across the entire attack lifecycle.

Packet replay techniques enable realistic simulation of network-based attack behaviors such as command-and-control communication, data exfiltration attempts, and other malicious traffic patterns. At the same time, sandbox-based malware execution allows organizations to evaluate how endpoint security controls respond to malicious code during the execution and impact phases of an attack.

By combining these approaches, it becomes possible to simulate a broader range of attacker behaviors and validate security controls across multiple stages of the attack lifecycle. This provides defenders with a more comprehensive understanding of how their defensive technologies perform when faced with real-world threat scenarios.

Conclusion

As cyber threats continue to evolve, organizations must move beyond static security assessments and develop practical methods for continuously validating the effectiveness of their defenses.

Traditional Breach and Attack Simulation approaches have provided valuable capabilities for testing security controls, but they also face important technical and operational challenges. Architectures that rely on vulnerable target hosts introduce security risks and operational complexity, while the inability to safely execute real malware limits the realism of simulated attack scenarios.

This paper has discussed two technical approaches that can help address these limitations. Packet replay techniques enable the simulation of network-based attack behaviors without requiring vulnerable target systems, while sandbox-based execution environments allow real malware samples to be executed safely within isolated environments.

By applying these techniques, organizations can conduct security validation in a manner that balances realism, safety, and operational practicality. Such approaches can help defenders gain deeper insight into how their security controls perform against real-world attack techniques, ultimately supporting more effective cyber defense strategies.

About digiDations

digiDations is an AI Native cybersecurity company helping organizations replace security assumptions with continuous, measurable validation. The question it answers is not whether an organization has vulnerabilities. It is whether their defenses can keep up with the threats targeting them right now.

Founded by cybersecurity veterans and AI researchers, digiDations built its product portfolio on Digital Mind, a proprietary AI foundation that continuously collects, correlates, and operationalizes real-world attack intelligence. The AI-Driven AEV portfolio includes ATLAS for Security Validation, APOLLO for Continuous Automated Red Teaming, and HELIOS for External Attack Surface Management. ORION for Threat Intelligence extends digiDations' CTEM coverage with adversary intelligence and visibility into emerging attack surfaces.

digiDations is building toward Autonomous Cyber Defense with TARA AI, the Autonomous CTEM Agent that self-directs across all four products: determining which scenarios to run, orchestrating execution, interpreting results, and generating reports, with teams approving decisions rather than defining the workflow.