VERODIN

VALIDATING VMWARE SERVICE-DEFINED FIREWALL EFFECTIVENESS WITH VERODIN

WHITEPAPER

vmware®

THE EVER-EVOLVING SECURITY CHALLENGE

Defenders are tasked with securing business-critical applications they don't operationally own or control. As evidenced by the OWASP Top 10, targeting application vulnerabilities has been a consistent and reliable vector for attackers. Rapid application development and the rising complexity of distributed and hybrid environments increase the difficulty of securing these applications exponentially. Defenders must adopt a mindset of assuming their organization will be breached and implement a focused and effective strategy to detect, isolate, and stop an attacker once a breach occurs.

Frameworks like MITRE ATT&CK™ are emerging as a reliable first step in categorizing attacker behaviors that defenders and defensive controls must learn to prevent, detect, and respond to as part of the "assume breach" mindset. By comparing controls and systems against attacker behaviors, organizations can establish an accurate baseline and prioritize infrastructure adjustments to gain better visibility and increase controls effectiveness.

Security professionals know that implementing advanced defenses like micro-segmentation and app control can dramatically increase the level of effort necessary for an attacker to be effective once an application has been breached but, historically, these approaches have been challenging to get right.

VMWARE SELECTS VERODIN TO VALIDATE EFFECTIVENESS

The VMware Service-Defined Firewall intrinsically embeds self-learning, adaptive micro-segmentation and app control, into the infrastructure, eliminating the need to bolt on additional products, deploy agents, or engage in complex configuration processes.

As VMware moves to increase the infrastructure's native security intelligence, it is important that customers and prospects alike can prove the effectiveness of the solution. Rather than making empty promises, VMware is committed to helping customers quantify the value of their investment and continuously validate that the Service-Defined Firewall is effectively deployed and configured in dynamic IT environments.

As the first step, VMware set out to ensure that its internal development processes are fully instrumented to enable the continuous validation of effectiveness for its solutions. After surveying the market, VMware selected Verodin's Security Instrumentation Platform (SIP) to instrument and validate the capabilities of the Service-Defined Firewall.

Verodin is the leader in enabling organizations to measure, manage, and improve their cybersecurity effectiveness. Customers operationalize Verodin SIP to validate that their security controls are effective, configured properly, and fully optimized on a continuous basis. Verodin SIP provides organizations with the evidence required to prove that their controls are actually delivering the desired protection for their business-critical assets.

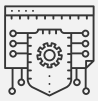
In this paper you will learn about the approach used in testing the VMware Service-Defined Firewall and the results of our tests. This paper will also provide a living use case for customers to be able to use Verodin in their own production environments as a means to prove their effectiveness.

TEST ENVIRONMENT

The test environment consisted of a traditional, three-tier web application deployed in vSphere with web (nginx), application (Drupal), and database (mySQL) tiers. VMware's Service-Defined Firewall (which leverages a powerful combination of the VMware NSX and VMware AppDefense Products) was deployed to provide the primary method of detection and control. Verodin SIP was also deployed into the environment and has three primary components in its architecture.

DIRECTOR

The Verodin SIP Director is the central management and reporting console. For this test, the Director was deployed in vSphere outside of the scope of the test application.



ACTORS

Verodin SIP Actors support multiple formats and are deployed into the environment to test endpoint, network, email, and cloud security controls. You can think of a Verodin SIP Actor as a software representation of a malicious threat actor. The environment was instrumented for effectiveness testing by deploying three Verodin SIP Actors: one in the application tier, one in the database tier, and one outside of the Service-defined Firewall's scope to represent a malicious threat actor outside of the datacenter.

INTEGRATIONS

The Verodin Director integrates into the various components of a customer's defensive stack in order to see how the controls prevent, detect, or miss executed tests. For this test, we configured Verodin's out-of-the-box integration with AppDefense. Additionally, NSX was configured to send its logs to an Elastic instance in the lab and Verodin's native Elastic integration was configured.

As the Verodin SIP Director instructs Actors to execute tests, it communicates with the defensive stack to pull data on what controls have visibility, what steps of the test are blocked, what detection events are created, where those events flow to, and ultimately if an actionable alert is generated. This process is referred to as the Verodin Effectiveness Validation Process ™ (EVP) and was co-developed between Verodin and number of leading organizations on the forefront of validating security effectiveness. By analyzing Verodin results, organizations can understand exactly how their controls and processes will perform before a breach occurs.

For more information on Verodin's architecture please go to **https://www.verodin.com/technology/platform**.

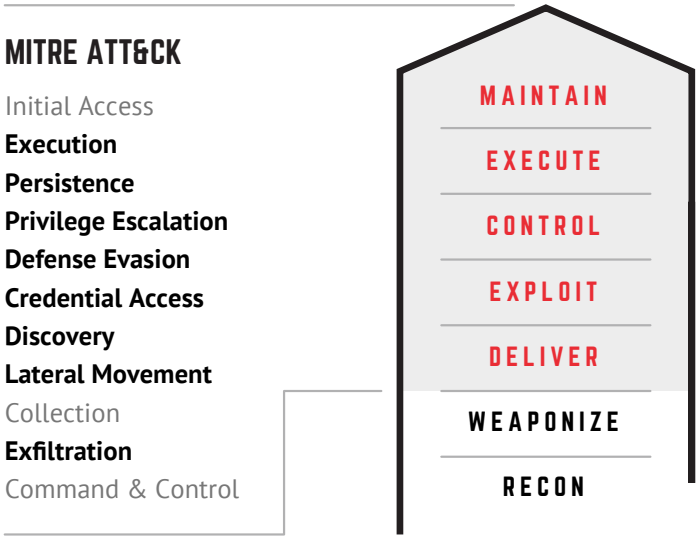
VERODIN TEST SEQUENCE

Verodin SIP contains a robust library of tests that the Verodin Behavior Research Team (BRT) updates on a consistent basis. Additionally, the library is extensible, enabling customers and partners to create tests themselves. These tests can even be shared within the Verodin community.

In Verodin SIP, individual control tests are called Actions. Actions can be chained together to form a Sequence representing several steps of the kill chain or a progression of tactics and techniques. Both Actions and Sequences are identified by a unique Verodin ID (VID).

Based on the configuration of the test environment, the Verodin SIP Sequence "Three tier app breach eight tactic progression ending in data exfil" (VID S100-096) was chosen. This Sequence was validated by Verodin BRT and designed as an "assume breach" use case for three tier applications. Assuming the breach of the application tier, the sequence executes several tactics before moving laterally and ultimately exfiltrating sensitive data from the database tier to an external actor.

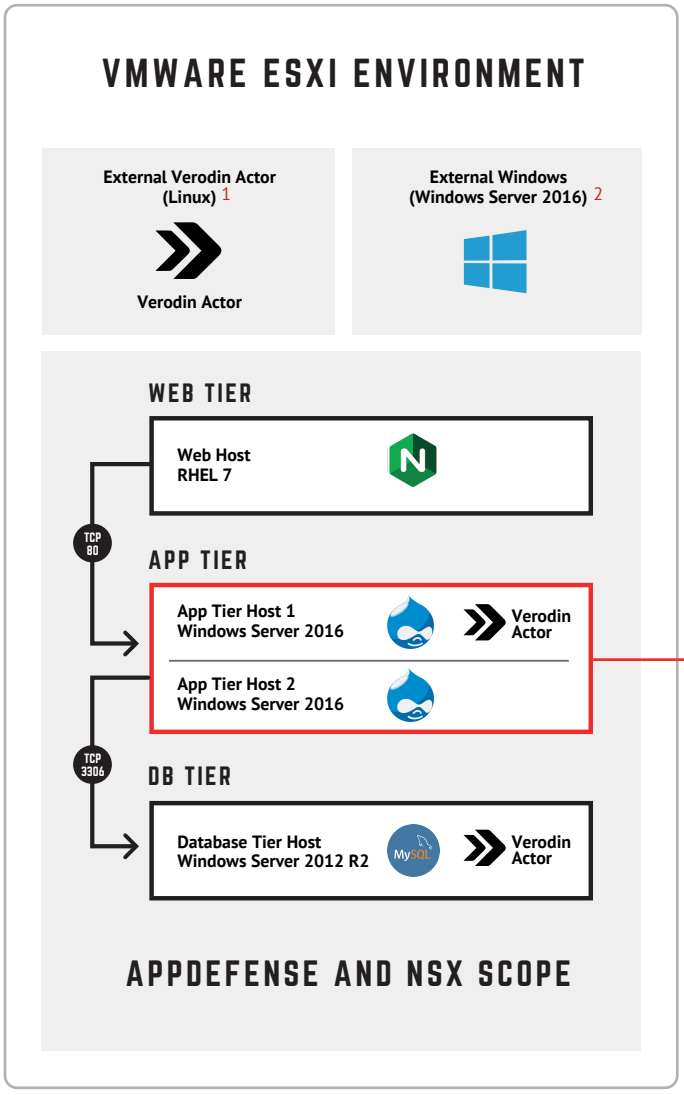
Through its progression, Sequence S100-096 executes techniques within eight of the eleven MITRE ATT&CK tactics, including: Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, and Exfiltration. The Sequence provides a realistic scenario to demonstrate the effectiveness of the Service-Defined Firewall's ability to dramatically increase the level of difficulty for an attacker to thrive post-breach.



MITRE ATT&CK is a framework of adversary tactics and techniques based on real-world observations. The Verodin test sequence used for this paper leverages techniques from the tactics in bold.

VMware ESXi Environment

There are three primary groups of tests within the Verodin Sequence S100-096.



1. Represents any untrusted system outside of AppDefense scope
2. Represents internal windows system outside of AppDefense scope

STEP 1

Initial actions by the Verodin Actor on the "breached" application tier:

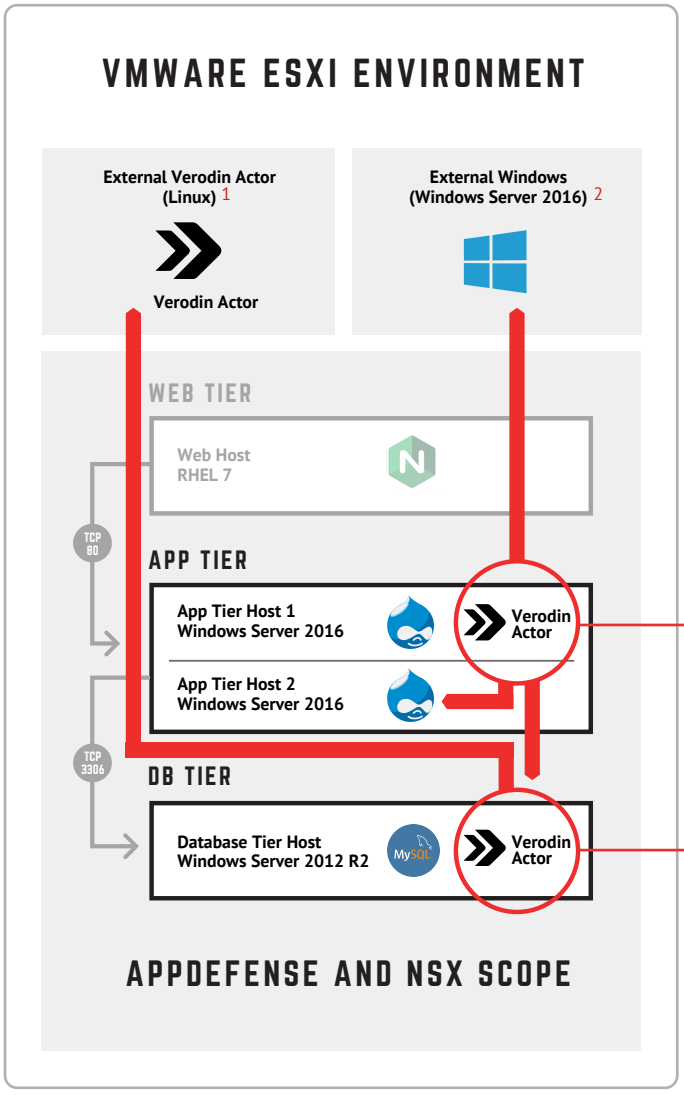
- Dumping credentials with Mimikatz (Credential Access) with both the standard binary as well as three custom binaries leveraging common defense evasion techniques:

  These evasions include executable padding, UPX packing, and a custom compilation with recognizable strings removed (Defense Evasion)

- Enumeration of local Administrators group memberships (Discovery)
- Bypassing network controls by downloading source code of a malicious tool and using native Windows tools to compile and execute locally (Execution, Defense Evasion)
- Creating a malicious process through rundll32 process execution (Execution, Defense Evasion)
- Creating a scheduled task for persistence (Persistence)
- Locally compiling a Windows service using native Windows tools and then executing the service to run with SYSTEM privileges (Defense Evasion, Persistence, Privilege Escalation, Execution)

VMware ESXi Environment

There are three primary groups of tests within the Verodin Sequence S100-096.



1. Represents any untrusted system outside of AppDefense scope
2. Represents internal windows system outside of AppDefense scope

STEP 2

Lateral movement actions from the Application tier to other systems

- Leveraging a mapped network drive, paexec, and Mimikatz to attempt to move laterally and dump credentials within the application tier and to the DB tier (Execution, Credential Access, Lateral Movement, Defense Evasion)

STEP 3

Establish persistence and perform unauthorized exfiltration of sensitive data from the DB tier to the external actor

- Creating a scheduled task for persistence (Persistence)
- Creating a malicious process through rundll32 process execution (Execution, Defense Evasion)
- From the Verodin Actor running on the DB tier Windows system, attempt to exfiltrate sensitive data out to the external Verodin SIP Actor both using the common HTTP tool curl and using native Powershell (Exfiltration, Execution)

TESTING PHASES

Verodin Sequence S100-096 was executed three times. For the first execution, the VMware Service-Defined Firewall was disabled in order to establish a baseline.

Next, VMware's Service-Defined Firewall self-learning was enabled and allowed to run in order to understand the application's normal behavior. This behavioral understanding of the application includes details on valid processes, how they are executed, and how they communicate over the network.

Once learning was complete, VMware NSX was enabled and configured to leverage the self-learned application understanding. Additionally, VMware AppDefense was turned on in Detect mode. In Detect mode, AppDefense will generate alerts for any unusual activity not identified as valid application behavior. After enabling these capabilities, the Verodin test sequence was executed again and the results were compared to the baseline.

Finally, AppDefense was changed to Prevent mode and set with a remediation action of block to any process or network activity that violated the application's known-good behavior. The Verodin sequence was run for a third time and results were again compared to both the previous run and baseline for final analysis.

NSX disabled
AppDefense disabled

NSX enabled
AppDefense enabled in detection mode

NSX enabled
AppDefense enabled in protection mode

BASELINE RESULTS WITH CONTROLS DISABLED

As expected, with all defensive controls turned off, the Verodin tests were able to execute successfully with nothing blocked or detected.

APP TIER

VID	Action Name	VERODIN TESTS FOR BASELINE DETECTED	VERODIN TESTS FOR BASELINE BLOCKED	NSX PREVENT/ DETECT, AD DETECT	AD/NSX PREVENT/ DETECT MODE
A104-167	Host CLI - Credential Access: Mimikatz (2.1.1)	●	●		
A104-166	Host CLI - Credential Access: Mimikatz W/ 10MB padding (2.1.1)	●	●		
A104-165	Host CLI - Credential Access, Defense Evasion: Mimikatz W/ UPX Packing (2.1.1)	●	●		
A104-059	Host CLI - Credential Access, Defense Evasion: Mimikatz W/ String Change	●	●		
A104-351	Host CLI - Discovery: Enumerate Local Administrators	●	●		
A104-218	Host CLI - Defense Evasion, Execution: RegAsm Bypass	●	●		
A104-096	Host CLI - Defense Evasion, Execution: rundll32.exe	●	●		
A104-010	Host CLI - Persistence: Scheduled Task	●	●		
A104-164	Host CLI - Defense Evasion, Execution, Persistence, Privilege Escalation: New Service	●	●		

APP TIER > APP TIER

VID	Action Name				
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	●	●		
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	●	●		
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	●	●		

APP TIER > EXTERNAL

VID	Action Name				
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	●	●		
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	●	●		
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	●	●		

APP TIER > DB

VID	Action Name				
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	●	●		
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	●	●		
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	●	●		

DB > EXTERNAL

VID	Action Name				
A104-010	Host CLI - Persistence: Scheduled Task	●	●		
A104-096	Host CLI - Defense Evasion, Execution: rundll32.exe	●	●		
A104-345	Host CLI - Execution, Exfiltration: HTTP Exfil/Upload of PCI Data using Powershell	●	●		
A104-344	Host CLI - Execution, Exfiltration: HTTP Exfil/Upload of PCI Data using Curl	●	●		

RESULTS WITH CONTROLS ENABLED IN DETECT MODE

After enabling NSX and AppDefense in Detect mode, Verodin Sequence S100-096 was re-run. As can be seen below, this had an immediate and significant impact compared to the baseline results. **100% of the actions were detected with meaningful alerts generated either by NSX and/or AppDefense**. Additionally, the lateral movement activities were blocked by NSX as it determined the network traffic did not match valid application behavior. This demonstrates a significant increase in defensive visibility and attacker isolation with a minimal configuration effort.

		VERODIN TESTS FOR BASELINE		NSX PREVENT/ DETECT, AD DETECT		AD/NSX PREVENT/ DETECT MODE
APP TIER		DETECTED	BLOCKED	DETECTED	BLOCKED	
VID	**Action Name**					
A104-167	Host CLI - Credential Access: Mimikatz (2.1.1)	🔴	🔴	🟢	🔴	
A104-166	Host CLI - Credential Access: Mimikatz W/ 10MB padding (2.1.1)	🔴	🔴	🟢	🔴	
A104-165	Host CLI - Credential Access, Defense Evasion: Mimikatz W/ UPX Packing (2.1.1)	🔴	🔴	🟢	🔴	
A104-059	Host CLI - Credential Access, Defense Evasion: Mimikatz W/ String Change	🔴	🔴	🟢	🔴	
A104-351	Host CLI - Discovery: Enumerate Local Administrators	🔴	🔴	🟢	🔴	
A104-218	Host CLI - Defense Evasion, Execution: RegAsm Bypass	🔴	🔴	🟢	🔴	
A104-096	Host CLI - Defense Evasion, Execution: rundll32.exe	🔴	🔴	🟢	🔴	
A104-010	Host CLI - Persistence: Scheduled Task	🔴	🔴	🟢	🔴	
A104-164	Host CLI - Defense Evasion, Execution, Persistence, Privilege Escalation: New Service	🔴	🔴	🟢	🔴	
APP TIER > APP TIER						
VID	**Action Name**					
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	🔴	🔴	🟢	🟢	
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	🔴	🔴	🟢	🟢	
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	🔴	🔴	🟢	🟢	
APP TIER > EXTERNAL						
VID	**Action Name**					
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	🔴	🔴	🟢	🟢	
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	🔴	🔴	🟢	🟢	
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	🔴	🔴	🟢	🟢	
APP TIER > DB						
VID	**Action Name**					
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	🔴	🔴	🟢	🟢	
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	🔴	🔴	🟢	🟢	
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	🔴	🔴	🟢	🟢	
DB > EXTERNAL						
VID	**Action Name**					
A104-010	Host CLI - Persistence: Scheduled Task	🔴	🔴	🟢	🔴	
A104-096	Host CLI - Defense Evasion, Execution: rundll32.exe	🔴	🔴	🟢	🔴	
A104-345	Host CLI - Execution, Exfiltration: HTTP Exfil/Upload of PCI Data using Powershell	🔴	🔴	🟢	🔴	
A104-344	Host CLI - Execution, Exfiltration: HTTP Exfil/Upload of PCI Data using Curl	🔴	🔴	🟢	🔴	

RESULTS WITH CONTROLS ENABLED IN PREVENT MODE

Before final execution of the Verodin SIP Sequence, AppDefense was configured to Prevent mode and instructed to not allow processes violating the learned application's known-good behavior to continue execution. As seen in the results below, this effectively eliminated the effectiveness of the attack vectors and behaviors used in the Verodin test sequence. 100% of the behaviors were both prevented and meaningful alerts were generated in both AppDefense and NSX.

		VERODIN TESTS FOR BASELINE		NSX PREVENT/ DETECT, AD DETECT		AD/NSX PREVENT/ DETECT MODE	
APP TIER		DETECTED	BLOCKED	DETECTED	BLOCKED	DETECTED	BLOCKED
VID	**Action Name**						
A104-167	Host CLI - Credential Access: Mimikatz (2.1.1)	●	●	●	●	●	●
A104-166	Host CLI - Credential Access: Mimikatz W/ 10MB padding (2.1.1)	●	●	●	●	●	●
A104-165	Host CLI - Credential Access, Defense Evasion: Mimikatz W/ UPX Packing (2.1.1)	●	●	●	●	●	●
A104-059	Host CLI - Credential Access, Defense Evasion: Mimikatz W/ String Change	●	●	●	●	●	●
A104-351	Host CLI - Discovery: Enumerate Local Administrators	●	●	●	●	●	●
A104-218	Host CLI - Defense Evasion, Execution: RegAsm Bypass	●	●	●	●	●	●
A104-096	Host CLI - Defense Evasion, Execution: rundll32.exe	●	●	●	●	●	●
A104-010	Host CLI - Persistence: Scheduled Task	●	●	●	●	●	●
A104-164	Host CLI - Defense Evasion, Execution, Persistence, Privilege Escalation: New Service	●	●	●	●	●	●
APP TIER > APP TIER							
VID	**Action Name**						
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	●	●	●	●	●	●
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	●	●	●	●	●	●
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	●	●	●	●	●	●
APP TIER > EXTERNAL							
VID	**Action Name**						
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	●	●	●	●	●	●
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	●	●	●	●	●	●
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	●	●	●	●	●	●
APP TIER > DB							
VID	**Action Name**						
A104-341	Host CLI - Lateral Movement: Copy Mimikatz using Mapped Network Drive	●	●	●	●	●	●
A104-342	Host CLI - Execution, Credential Access: Remote Execution of Mimikatz using PaExec	●	●	●	●	●	●
A104-343	Host CLI - Defense Evasion: Removal of Network Share Connection	●	●	●	●	●	●
DB > EXTERNAL							
VID	**Action Name**						
A104-010	Host CLI - Persistence: Scheduled Task	●	●	●	●	●	●
A104-096	Host CLI - Defense Evasion, Execution: rundll32.exe	●	●	●	●	●	●
A104-345	Host CLI - Execution, Exfiltration: HTTP Exfil/Upload of PCI Data using Powershell	●	●	●	●	●	●
A104-344	Host CLI - Execution, Exfiltration: HTTP Exfil/Upload of PCI Data using Curl	●	●	●	●	●	●

SUMMARY

These tests performed using Verodin SIP demonstrate the VMware Service-Defined Firewall's ability to reduce the attack surface with minimal effort. Common attacker tactics and techniques become increasingly difficult to execute when the infrastructure itself is enforcing known-good application behavior and communications.

Additionally, it is important to note other differentiators for the VMware Service-Defined Firewall:

Ease of deployment

AppDefense provides process-level detection and control with no agents to deploy. At the network level, there are no span ports, taps, or "inline" issues to manage. Removing these requirements reduces complexity and greatly decreases the ability for changes in IT production infrastructures to create "environmental drift" which can cripple control effectiveness.

Seamless visibility and coverage

When the infrastructure fabric is acting as an intelligent security control, challenges of control visibility and coverage are no longer relevant.

Significate reduction in time and effort to configure

Intelligent self-learning dramatically reduces the need for manually intensive, complex configuration.

Wide attack coverage and attack surface reduction with centralized control

VMware's Service-Defined Firewall provides both a combination of endpoint and network controls through its embedded self-learning, adaptive micro-segmentation and deep application control. Additionally, as demonstrated by the Verodin test sequence results, several common endpoint and network control evasion techniques are rendered useless, limiting the scope of where defenders need to focus.

This test was designed to focus on validating the effectiveness of the specific controls provided by the VMware Service-Defined Firewall. The selected tests are elemental to many attacks, but customers should leverage the complete Verodin library to validate their defensive stacks against a comprehensive set of behaviors. Certain attacker techniques, such as process injection, were out of scope for this test and should be included in any customer test.

That being said, the dramatic reduction in attack surface that the Service-Defined Firewall delivers, combined with the other benefits described above, enables defenders to laser focus on that specific vector and significantly elevate the bar of skill needed for an attacker to be successful. VMware's Service-Defined Firewall is an important platform for any organization embracing the "assume breach" mindset and seeking ways to reduce the ability for attackers to thrive in their environment.

VMware customers and prospects should leverage Verodin to assess current control gaps, quantify the value of adopting the Service-Defined firewall, and continuously validate that their controls are effectively deployed and configured in dynamic IT environments.

Verodin

Interested in using Verodin SIP to test and validate your own environment? To learn more, please visit:

https://www.verodin.com.

VMware

To learn more about VMWare's Service-defined firewall, please visit:

vmware.com/go/service-defined-firewall.

VERODIN

Verodin's Security Instrumentation Platform (SIP) provides evidence of the effectiveness of customers' cybersecurity controls, enabling them to validate the protection of their business-critical assets. Verodin has a diverse, global customer base and is backed by world-class investors including Bessemer Venture Partners, Blackstone, Capital One Growth Ventures, Cisco Investments, Citi Ventures, ClearSky, Crosslink Capital, Rally Ventures and TenEleven Ventures. To learn more about Verodin, please visit www.verodin.com.

APPENDIX

To supplement the detailed evaluation report, this section contains screenshots collected by the evaluation team.

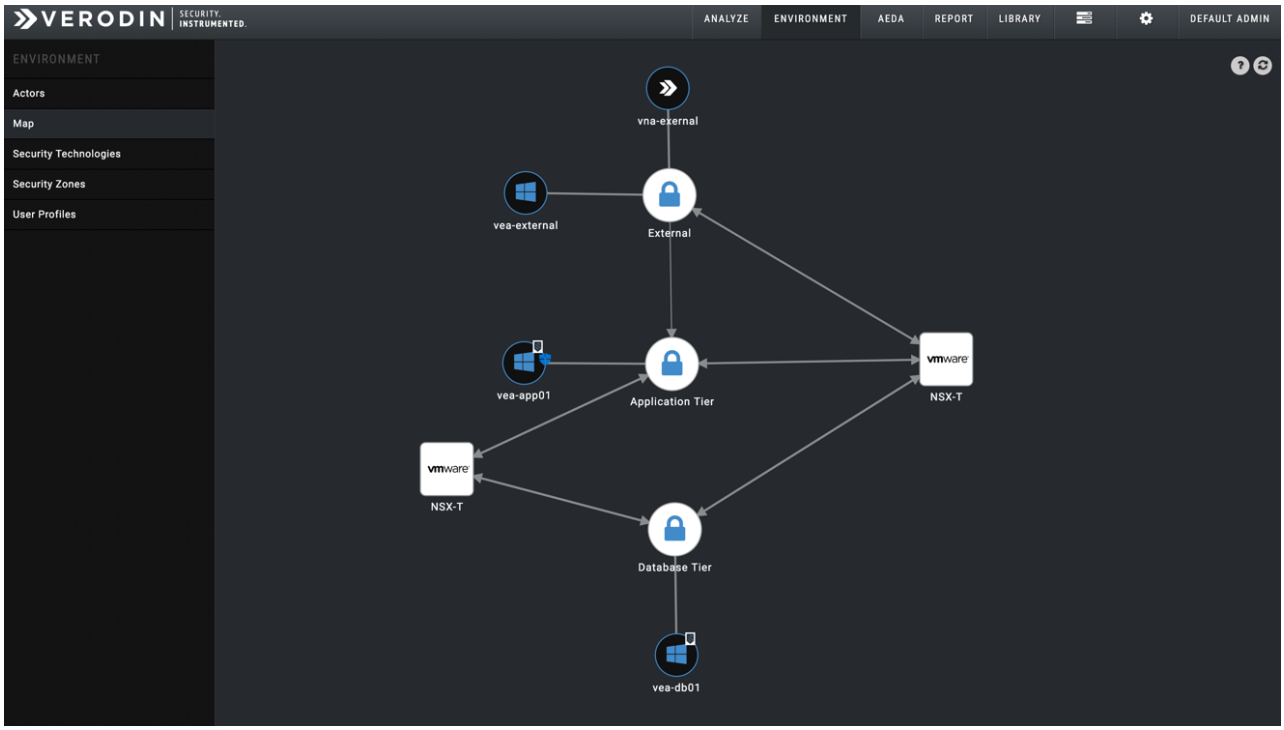


Figure 1: Verodin SIP Director displaying a map of the target three-tier web application deployed in vSphere.

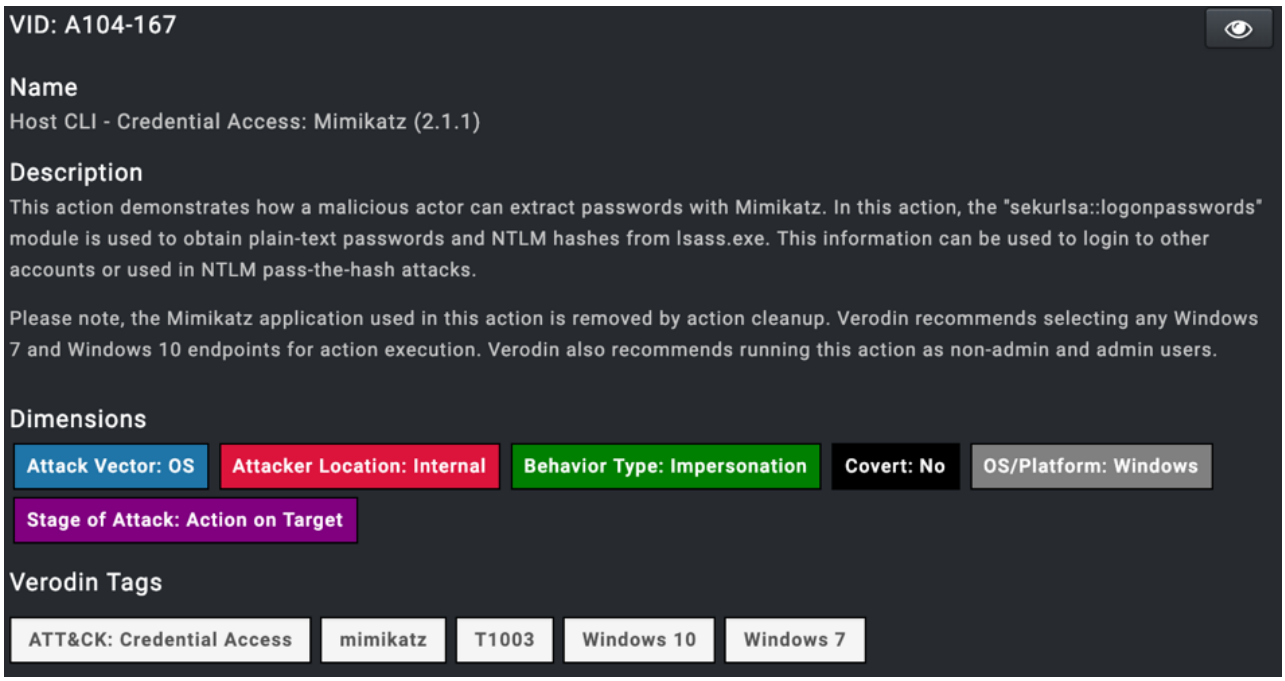


Figure 2: Verodin SIP Director displaying the details of an individual control test, called an Action. For this evaluation, actions were chained together to form a progression of tactics and techniques.

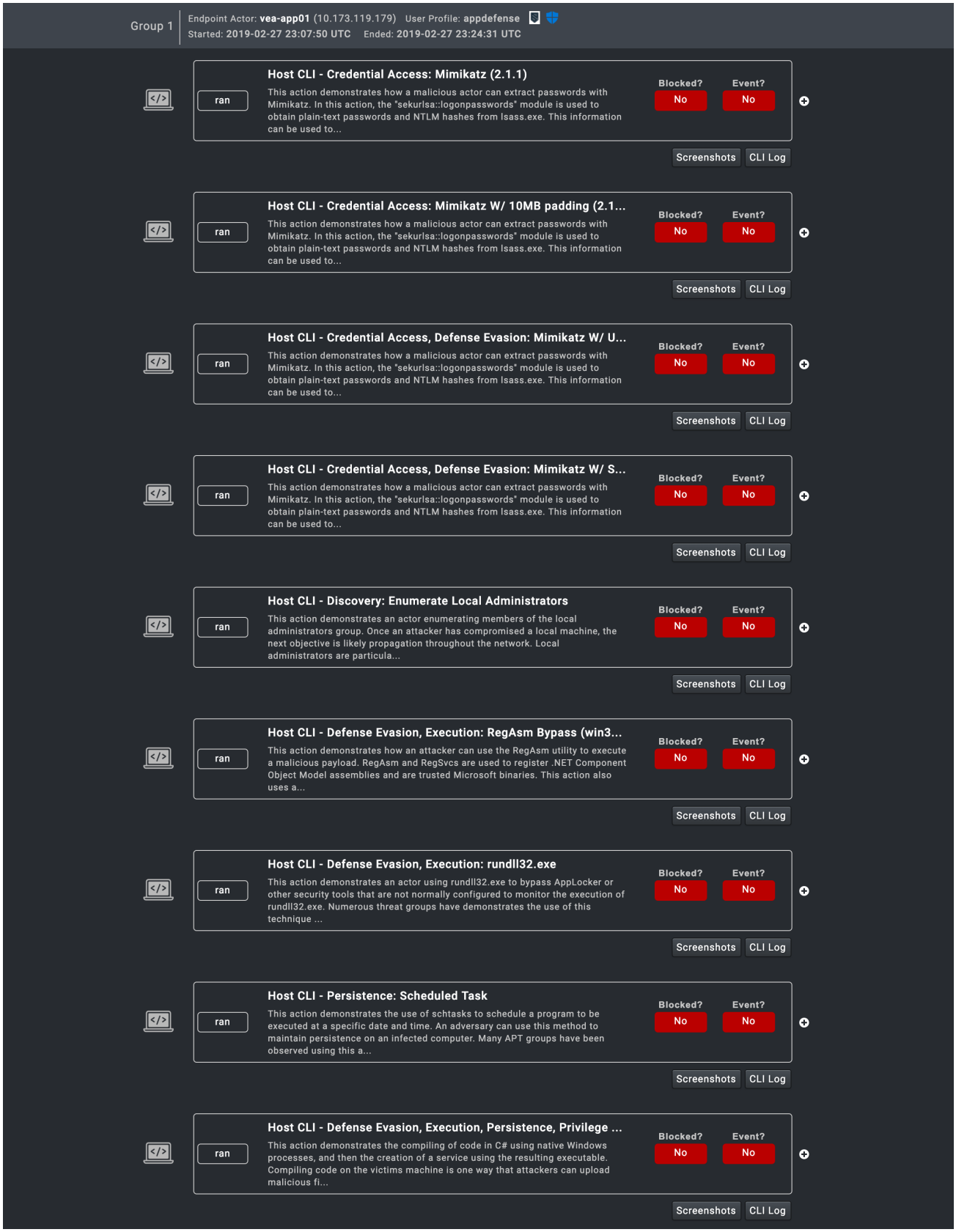
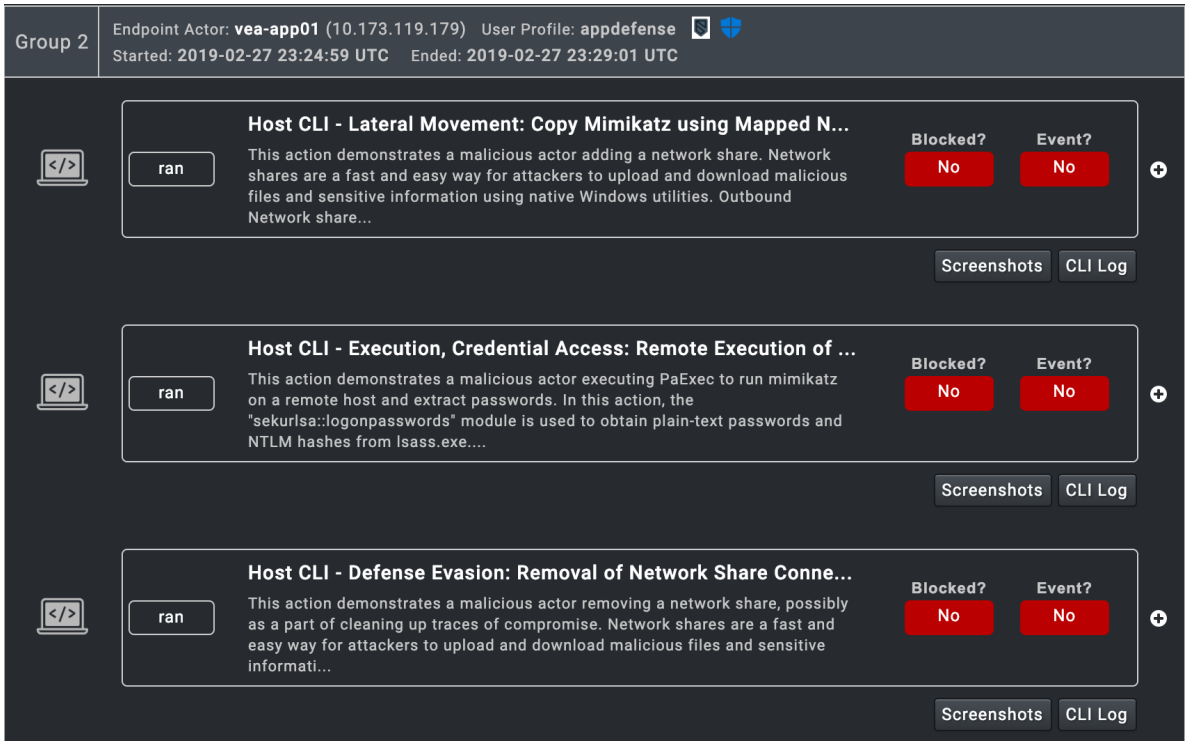
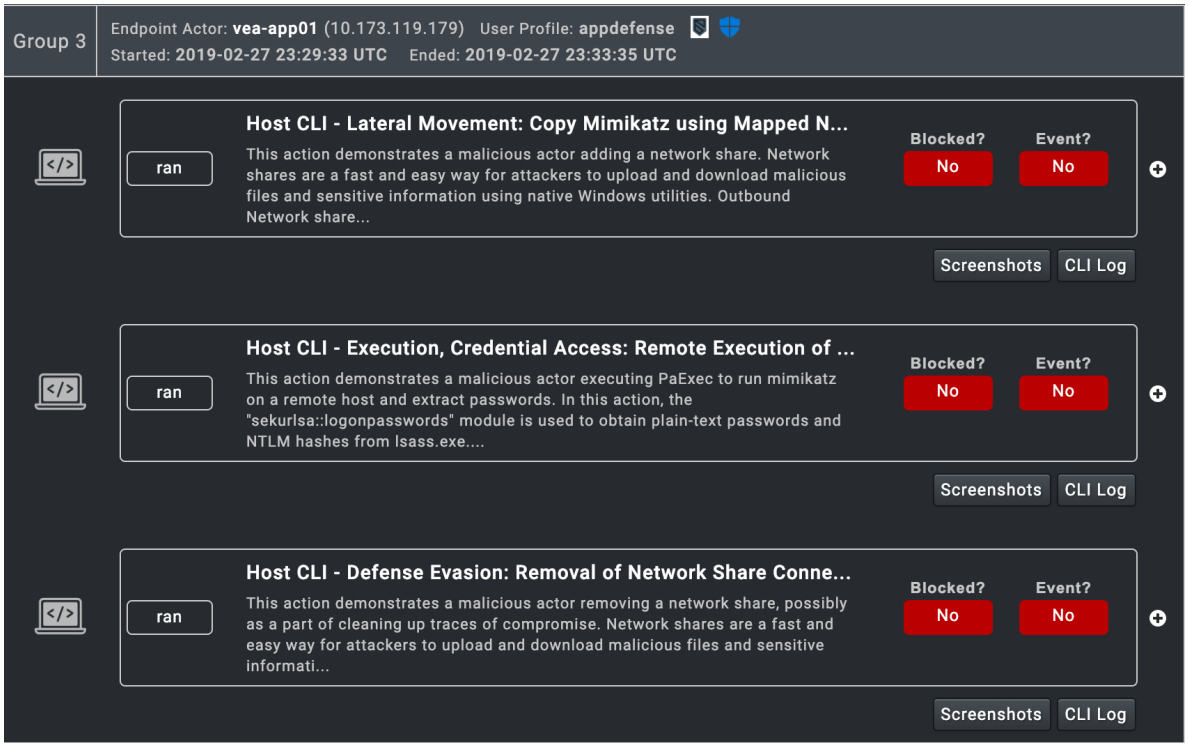


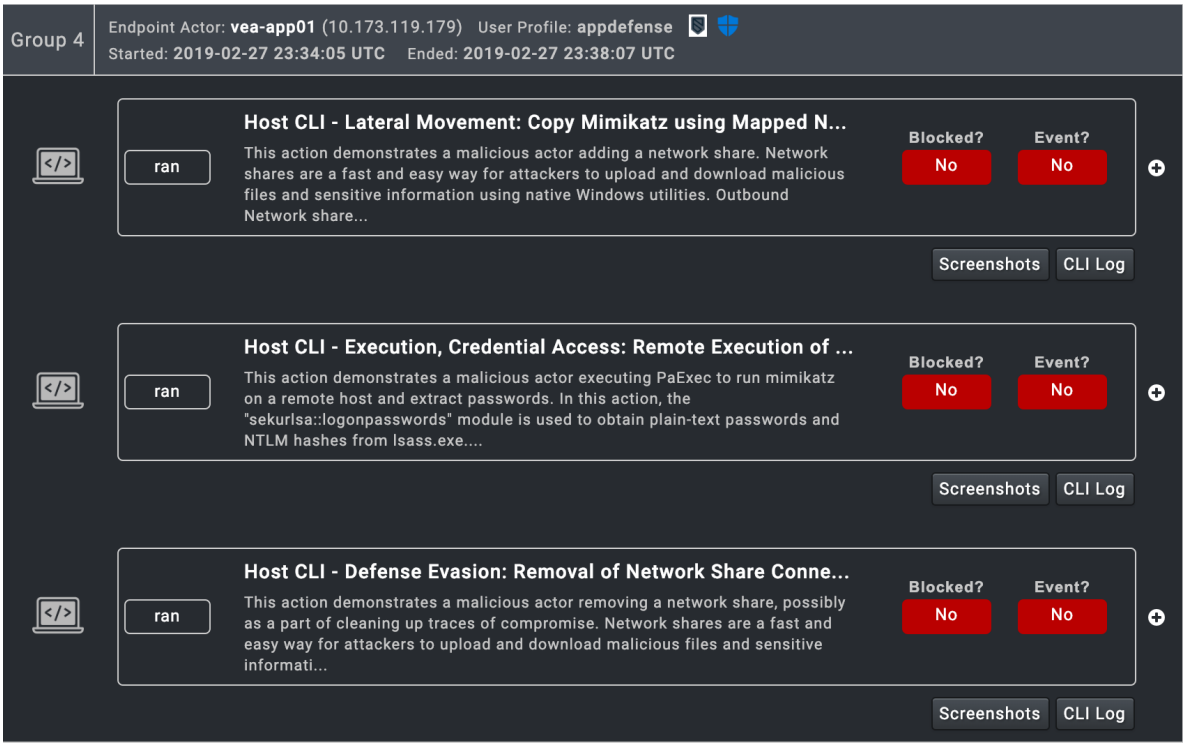
Figure 3: Base configuration: Verodin SIP results showing an attacker's ability to access credentials, evade defenses, enumerate devices, and gain persistence, without being detected or prevented, once on the application server.



Figures 4: Base configuration: Verodin SIP results showing an attacker's ability to move laterally to another application server, access additional credentials, and remove network shares, once on the application server.



Figures 5: Base configuration: Verodin SIP results showing an attacker's ability to move laterally to external systems, access additional credentials, and remove network shares, once on the application server.



Figures 6: Base configuration: Verodin SIP results showing an attacker's ability to move laterally to database systems, access additional credentials, and remove network shares, once on the application server.

Figure 7: Base Configuration: Verodin SIP results showing an attacker's ability to gain persistence, evade defenses, and exfiltrate PCI data, once on the database server.

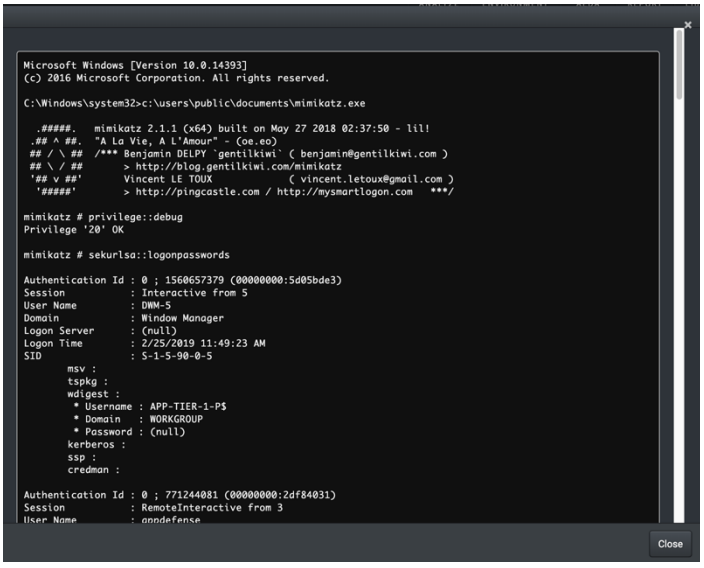


Figure 8: Base Configuration: Verodin SIP Director displaying command line output resulting from successful execution of Action A104-167, credential access, against the target environment.

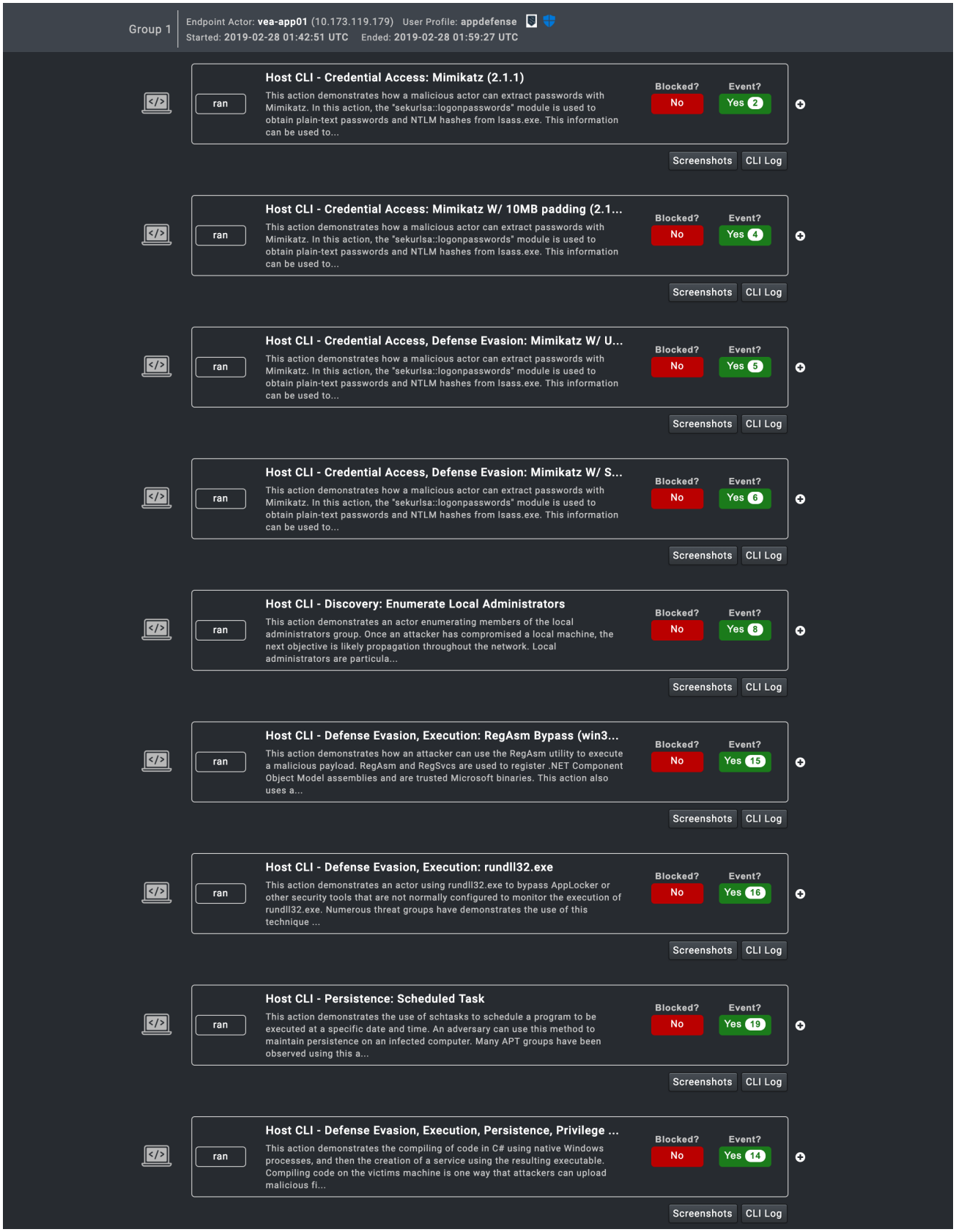


Figure 9: NSX Prevent/Detect, AD Detect Mode: Verodin SIP Director validating that VMware service-defined firewall detects the attacker's attempts to access credentials, evade defenses, enumerate devices, and gain persistence, once on the application server.

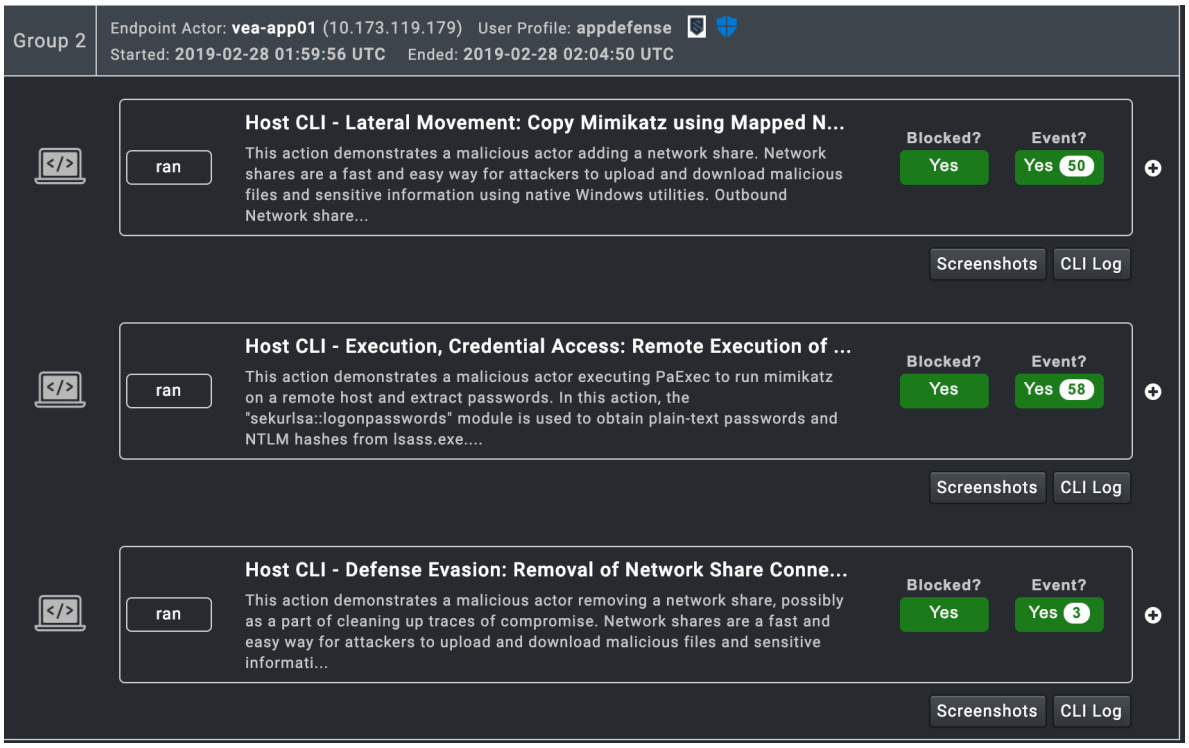


Figure 10: NSX Prevent/Detect, AD Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to move laterally to another application server, access additional credentials, and remove network shares, once on the application server.

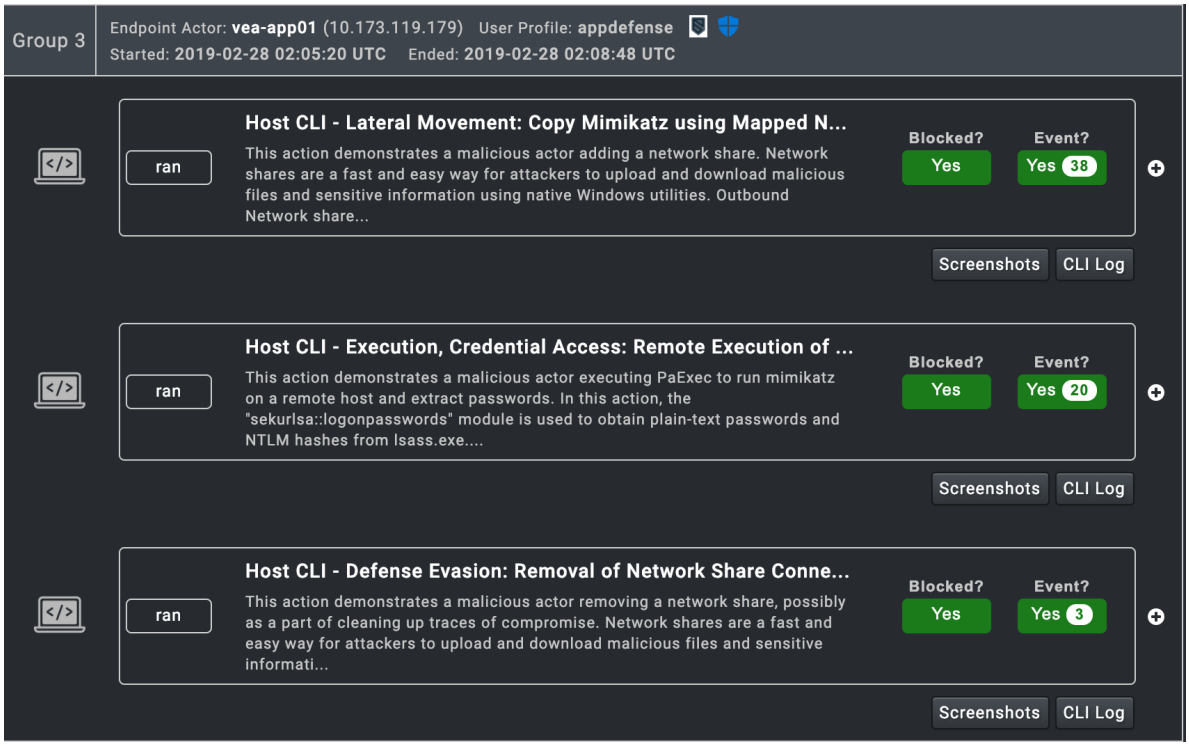


Figure 11: NSX Prevent/Detect, AD Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to move laterally to external systems, access additional credentials, and remove network shares, once on the application server.

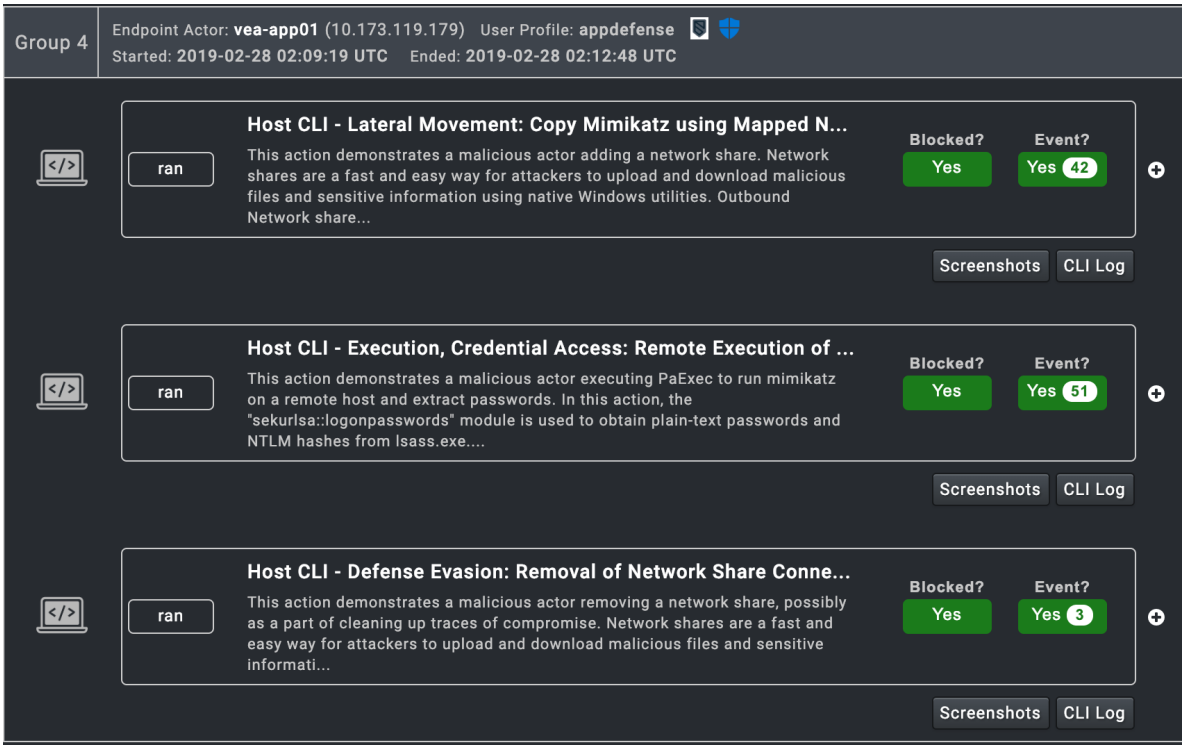


Figure 12: NSX Prevent/Detect, AD Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to move laterally to the database server, access additional credentials, and remove network shares, once on the application server.

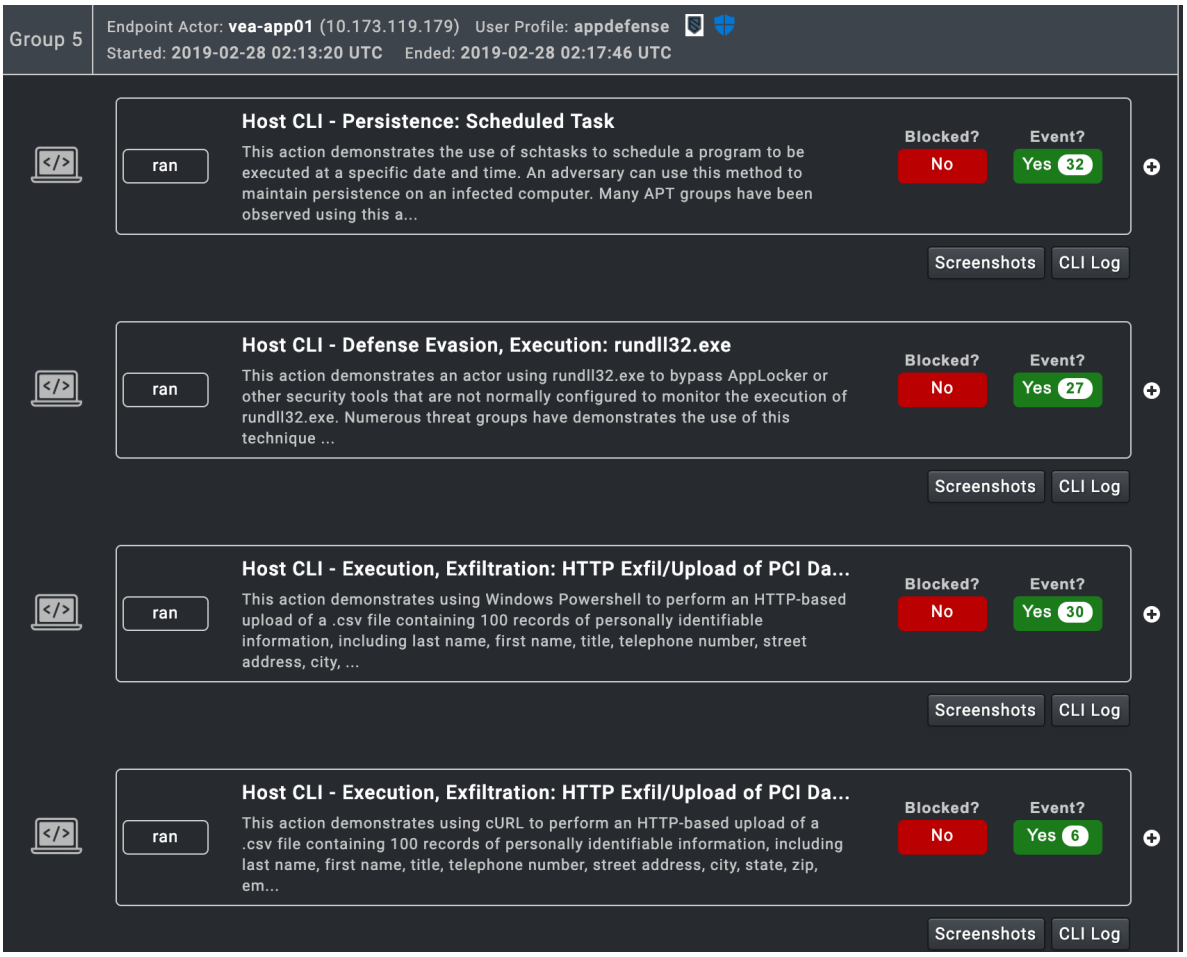


Figure 13: NSX Prevent/Detect, AD Detect Mode: Verodin SIP Director validating that VMware service-defined firewall detects the attacker's attempts to gain persistence, evade defenses, and exfiltrate PCI data, once on the database server.

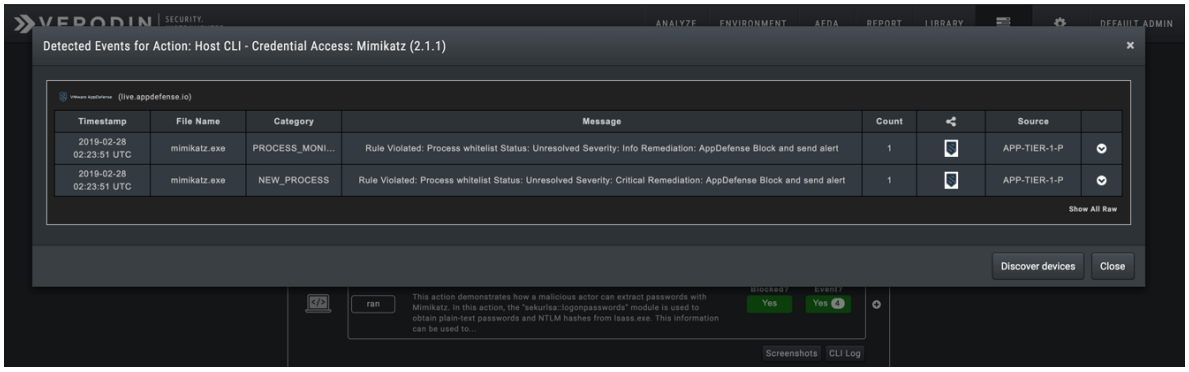


Figure 14: NSX Prevent/Detect, AD Detect Mode: Leveraging Verodin's native integrations with VMware AppDefense and Elastic, Verodin SIP director displaying successful detection of credential access.

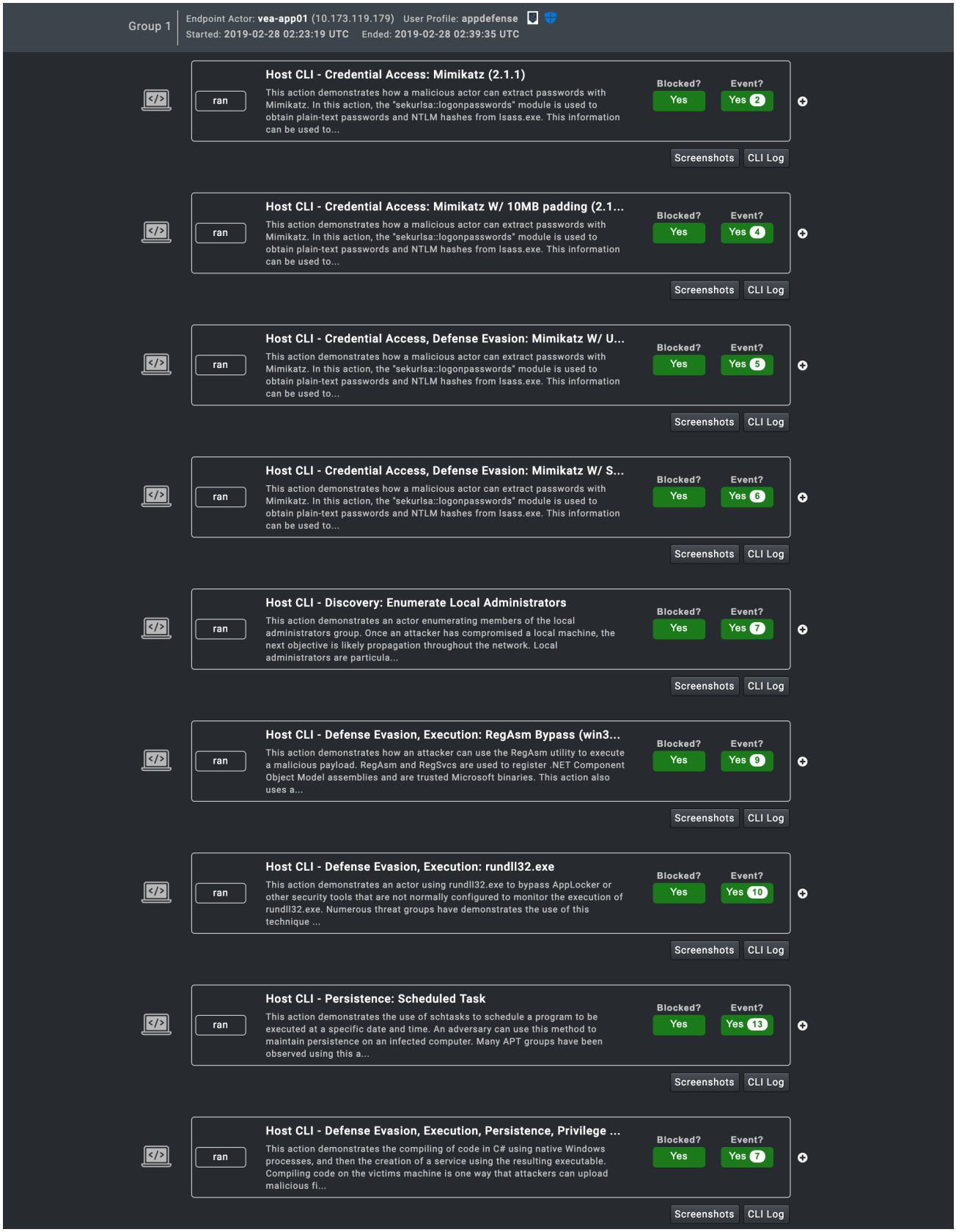


Figure 15: AD/NSX Prevent/ Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to access credentials, evade defenses, enumerate devices, and gain persistence, once on the application server.

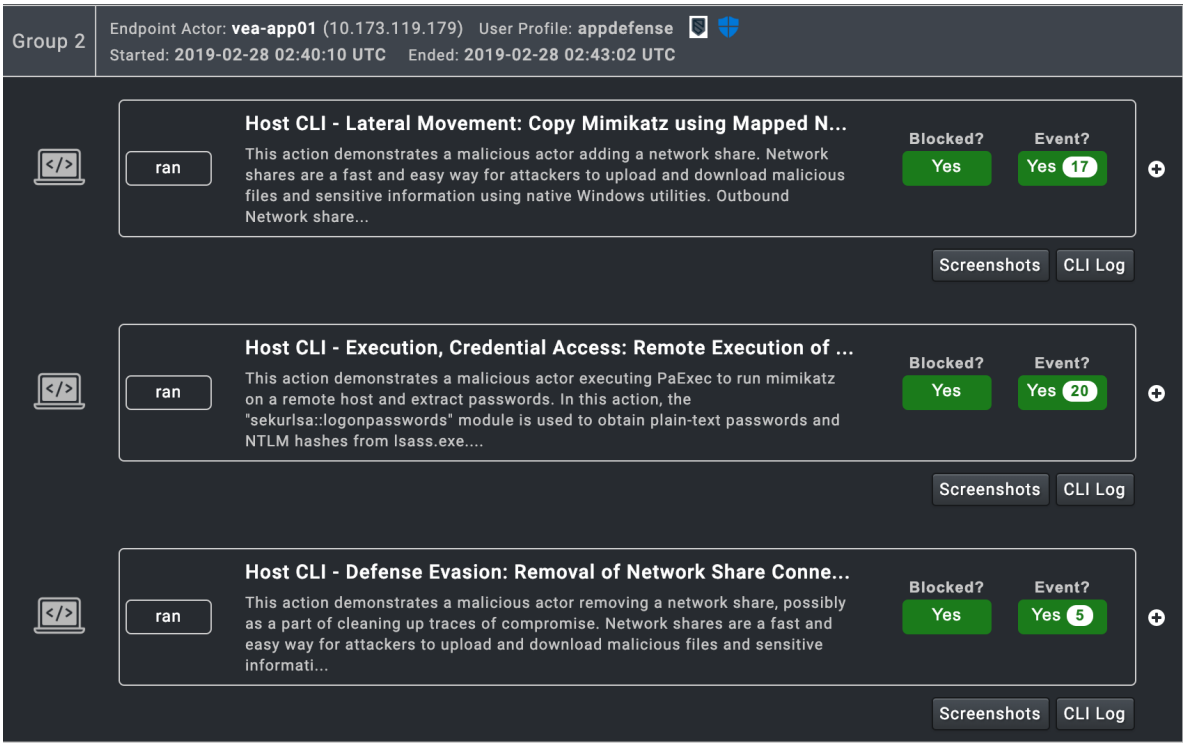


Figure 16: AD/NSX Prevent/ Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to move laterally to another application server, access additional credentials, and remove network shares, once on the application server.

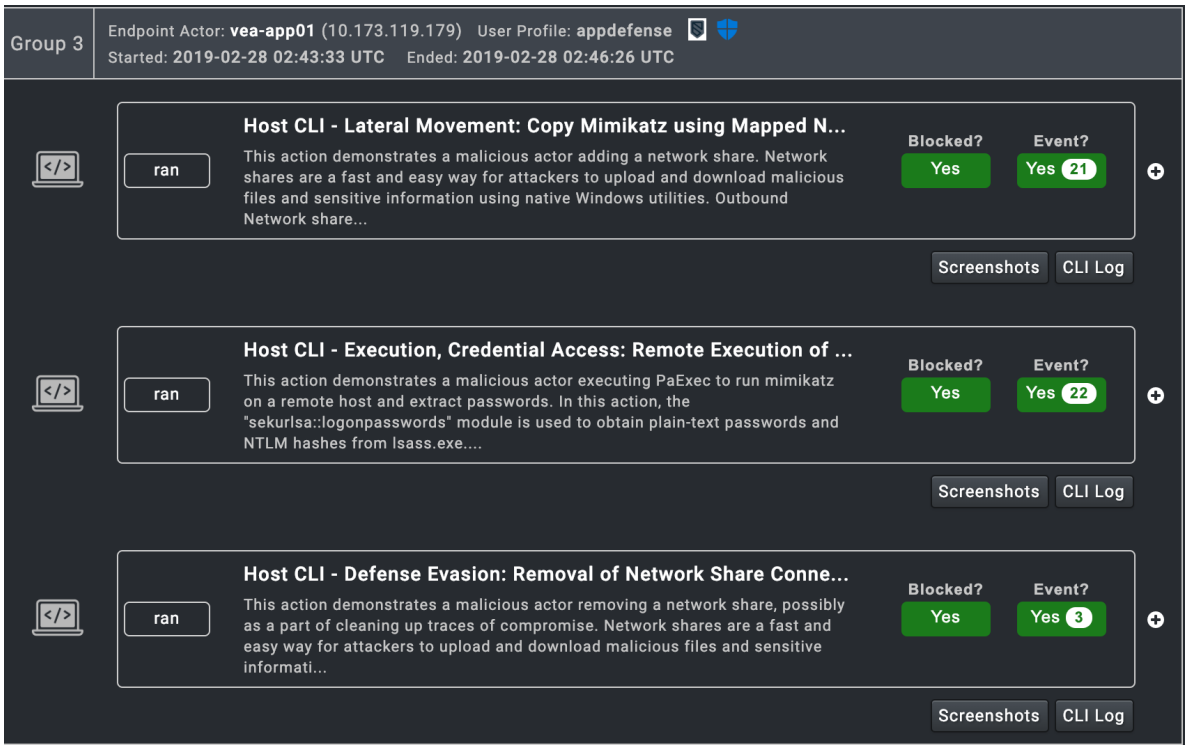


Figure 17: AD/NSX Prevent/ Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to move laterally to external systems, access additional credentials, and remove network shares, once on the application server.

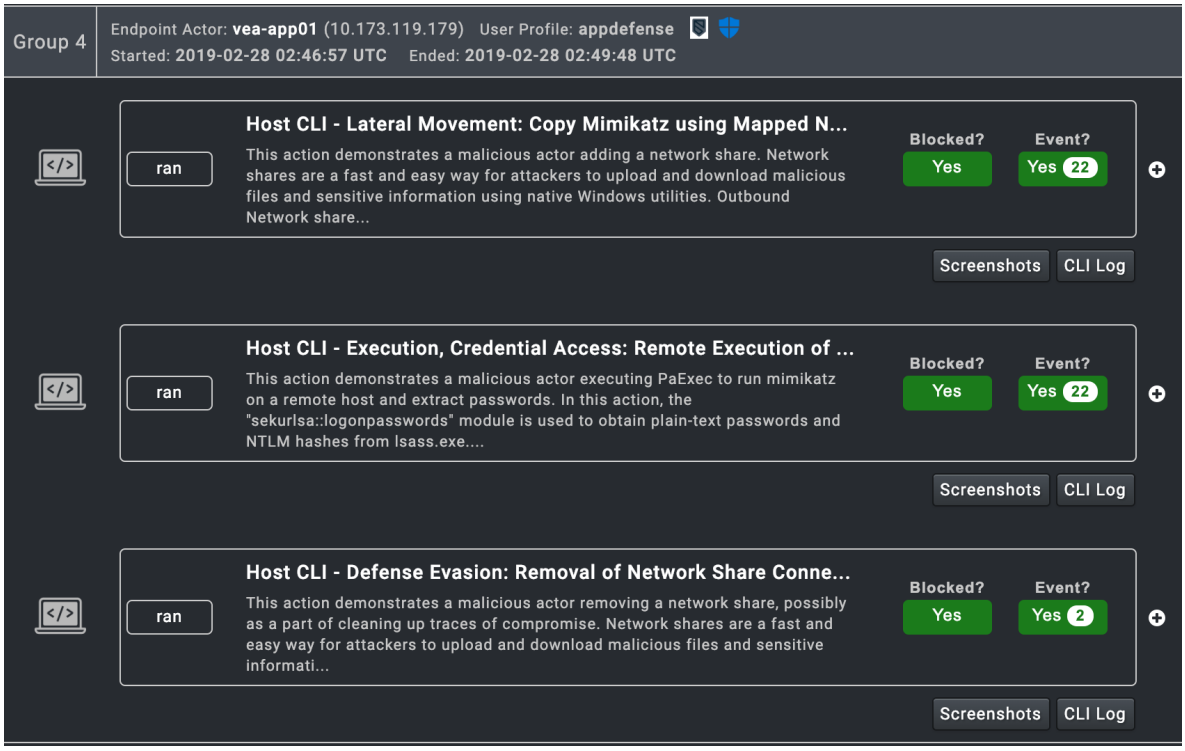


Figure 18: AD/NSX Prevent/ Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to move laterally to the database server, access additional credentials, and remove network shares, once on the application server.

Figure 19: AD/NSX Prevent/ Detect Mode: Verodin SIP Director validating that VMware service-defined firewall prevents and detects the attacker's attempts to gain persistence, evade defenses, and exfiltrate PCI data, once on the database server.

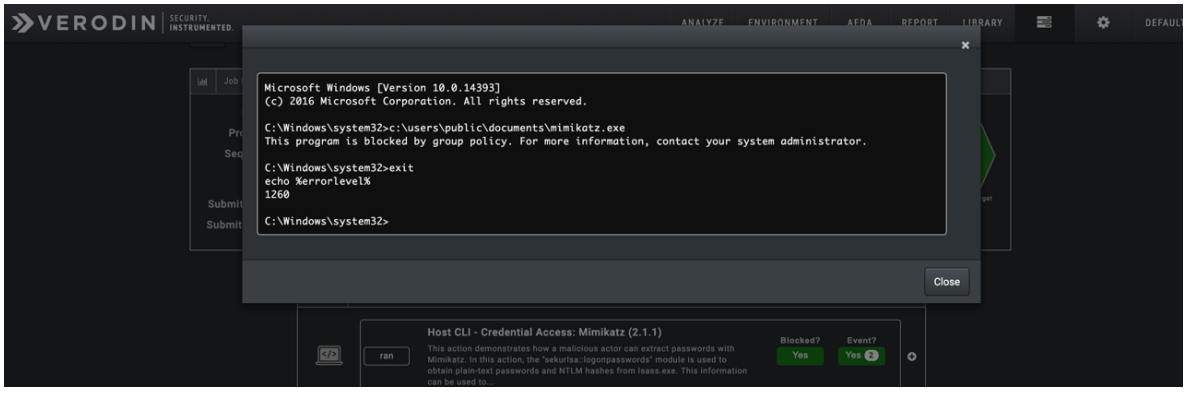


Figure 20: AD/NSX Prevent/ Detect Mode: Verodin SIP Director displaying blocked credential theft attempt, once VMware NSX was enabled, training was complete, and VMware AppDefense was turned on in Prevent mode.