Infiltrating Defenses: Abusing VMware in MITRE's Cyber Intrusion

By Lex Crumpton

Published: 2025-04-03 · Archived: 2026-04-05 14:48:41 UTC

Written by

and Charles Clancy.

⚠️ **This post has moved to our new blog and is kept here for archival purposes.**
https://ctid.mitre.org/blog/2024/05/22/infiltrating-defenses-abusing-vmware-in-mitres-cyber-intrusion/

Press enter or click to view image in full size

Image Credit: GPT-4o / DALL-E 3

This is the third and final blog post in a series detailing MITRE's encounter with a state-sponsored cyber threat actor in our research and experimentation network, NERVE. It builds upon the insights shared in our April 19, 2024 post, "Advanced Cyber Threats Impact Even the Most Prepared" and May 3, 2024 post "Technical Deep Dive: Understanding the Anatomy of a Cyber Intrusion". We continue to work across MITRE, including our Information Security Team, to help all security teams understand and defend against this threat.

In this post of our series, we provide technical details of new behavior employed by the adversary, who aligns with Google Mandiant's UNC5221, and how the **BRICKSTORM** backdoor and **BEEFLUSH** web shell abused VMs in VMware through a privileged user account, VPXUSER, to establish persistence within the impacted environment. We will also provide detection scripts, from MITRE and CrowdStrike, to find this activity in other environments and go over how Secure Boot serves as a barrier against the adversary technique.

Recap from Parts One & Two

In our first blog post, we shared the experience of facing a cyber intrusion that targeted MITRE's Networked Experimentation, Research, and Virtualization Environment (NERVE) through two Ivanti Connect Secure zero-

day vulnerabilities that bypassed our multi-factor authentication. The adversary maneuvered within the research network via VMware infrastructure using a compromised administrator account, then employed a combination of backdoors and web shells to maintain persistence and harvest credentials.

In our second blog post, we took a deep dive into the technical details of the intrusion, including a timeline of events, indicators of compromise, and malware analysis. Additionally, we disclosed novel aspects not previously reported by Mandiant or other threat intelligence sources, including:

- Details on the **BEEFLUSH** web shell; and
- Unique components of the **BUSHWALK** web shell seen in our incident.

Table 1. Notable MITRE ATT&CK® techniques shared in our initial blog

Before delving into the techniques employed by the adversary to abuse VMware infrastructure, it is essential to understand the overarching context: the adversary had already gained administrative access to NERVE ESXi infrastructure.

Created Rogue VMs

Rogue VMs are created and managed through service accounts directly on the hypervisor, rather than through the vCenter administrative console. As a result, these VMs do not appear in the inventory.

As we said in the second post, “On January 5, 2024, the adversary escalated their attack with manipulated VMs and compromised administrative credentials to establish control over the infrastructure. Specifically, their actions included attempted enablement of SSH, destruction of one of their own VMs, and file downloads.”

The adversary created their own rogue VMs within the VMware environment, leveraging compromised vCenter Server access. They wrote and deployed a JSP web shell (**BEEFLUSH**) under the vCenter Server’s Tomcat server to execute a Python-based tunneling tool, facilitating SSH connections between adversary-created VMs and the ESXi hypervisor infrastructure.

> By deploying rogue VMs, adversaries can evade detection by hiding their activities from centralized management interfaces like vCenter. This allows them to maintain control over compromised systems while minimizing the risk of discovery.

Detecting Adversary Activity in VMware Ecosystem

In VMware’s environment, spotting adversary activity demands meticulous scrutiny. For instance, adversaries might enable SSH on hypervisors and log in by routing traffic through the vCenter Server. This technique underscores the importance of monitoring SSH activity for signs of unauthorized access.

What to Look for:

1. **Anomalous SSH Enablement:** Keep a close watch for unexpected occurrences of “SSH login enabled” messages. Any activation of SSH outside the normal administrative cycle could indicate malicious activity.

2. **Unusual SSH Sessions**: Monitor for deviations from the expected pattern of SSH sessions being opened. Look out for instances where “SSH session was opened for” messages occur unexpectedly or at unusual times.

Table 2. Notable ATT&CK techniques

Deploying Rogue VMs

As we said in the second post, “Moving forward to January 7, 2024, the adversary accessed VMs and deployed malicious payloads, the **BRICKSTORM** backdoor and the **BEEFLUSH** web shell. The adversary also used a default VMware account, VPXUSER, to make seven API calls that enumerated a list of mounted and unmounted drives.”

The adversary bypassed detection mechanisms by deploying **rogue** VMs, as VPXUSER, directly onto hypervisors using SFTP to write files then executed them with `/bin/vmx`. By doing this, these rogue VMs were not discoverable via vCenter, the ESXi web interface, and even some on-hypervisor command-line utilities that query the API.

These rogue VMs contained the **BRICKSTORM** backdoor and persistence mechanisms, see blog post 2, that were configured with dual network interfaces for communication with both the Internet/C2 and core administrative subnets within the prototyping network.

> Adversaries often can leverage the vpxuser account to perform various administrative tasks, such as enumerating VMs, accessing configuration settings, and interacting with the underlying hypervisor infrastructure. Additionally, adversaries may deploy rogue VMs directly onto hypervisors to evade detection mechanisms and maintain persistence within the environment. Rogue VMs, which are created and operated without proper authorization and management by the hypervisor, provide adversaries with a stealthy foothold for conducting malicious activities. These VMs can bypass visibility controls within VMware management interfaces, making them difficult to detect and mitigate.

Detecting Rogue VMs

Safeguarding against rogue VMs and any ensuing persistence demands a vigilant approach. Simply using the hypervisor management interface to manage VMs is often insufficient and can be pointless when it comes to dealing with rogue VMs. This is because rogue VMs operate outside the standard management processes and do not adhere to established security policies, making them difficult to detect and manage through the GUI alone. Instead, one needs special tools or techniques to identify and mitigate the risks associated with rogue VMs effectively.

What to Look For:

Command-Line Usage: Utilize the following commands on an ESXi hypervisor to identify unregistered VMs:

1. `vim-cmd vmsvc/getallvms`
2. `esxcli vm process list | grep Display`

Comparison of VM Lists: Compare the output of `vim-cmd` (API-based VM check) with the list of running VMs obtained from `esxcli`.

1. Differences in the list of VMs between the output of a `vim-cmd` (that will check for VMs via the API) and the list of running VMs that `esxcli` sees (which directly queries the host hypervisor) indicate a potential problem. A VM running on a hypervisor that is not seen via the registered VM data via API warrants further investigation as a possible unregistered/rogue VM.

Detecting VMware Persistence

To address the persistence of these rogue VMs, it is crucial to scrutinize the hypervisor's startup scripts.

What to Look For

Persistence Mechanism: Monitor for modification of the legitimate `/etc/rc.local.d/local.sh` file to include the following line:

```
/bin/vmx -x /vmfs/volumes/<REDACTED_VOLUME>/<REDACTED_VM_NAME>/<REDACTED_VM_NAME>.vmx 2>/dev/null 0>
```

Persistence Identification: Search for invocations of the `/bin/vmx` binary within `/etc/rc.local.d/` or more specifically by manually reviewing the `local.sh` startup script with the following commands:

1. `grep -r \/bin\/vmx /etc/rc.local.d/`
2. `cat /etc/rc.local.d/local.sh`

Table 3. Notable ATT&CK techniques

Suspicious VMware Detection Scripts

MITRE is sharing two scripts designed to identify and mitigate potential threats within the VMware environment. The first script, developed by MITRE, Invoke-HiddenVMQuery is written in PowerShell and serves to detect malicious activities. It scans for anomalous invocations of the `/bin/vmx` binary within `rc.local.d` scripts.

Furthermore, it checks for the presence of rogue VMs by cross-referencing data obtained from two sources: the `vim-cmd` utility, which queries VMs via the API, and the list of running VMs retrieved by `esxcli`, a command-line interface directly querying the host hypervisor. Any VM detected running on a hypervisor but not listed via the registered VM data via API warrants immediate investigation as a potential threat.

We are also sharing a PowerShell script, VirtualGHOST, that CrowdStrike prepared to help detect evidence of unregistered VMs using PowerCLI and help scale hunting exercises by executing the scripts remotely. Thanks to CrowdStrike for their collaboration in identifying these adversary tactics, techniques, and procedures (TTPs).

By leveraging these scripts, organizations can identify and respond to suspicious activities, bolstering their cybersecurity defenses against evolving threats.

Recommended Mitigation Strategy

Based on consultation with the VMware PSIRT team, the most effective countermeasure to thwart the persistence mechanism is to enable secure boot. Secure boot is a security feature designed to verify the integrity of a host's boot process, mitigating the risk of unauthorized modifications.

Enabling secure boot serves as one defense against adversaries seeking to establish persistent access within the VMware environment. By verifying the integrity of the boot process, secure boot prevents malicious actors from injecting unauthorized code.

This countermeasure aligns with the MITRE ATT&CK Mitigation: Boot Integrity (M1046).

By fortifying the boot process with secure boot, organizations can thwart adversaries' efforts to evade detection and maintain unauthorized access to critical systems.

For detailed information on this feature and its implementation, please refer to the following resource: VMware Secure Boot Documentation.

Conclusion

As adversaries continue to evolve their tactics and techniques, it is imperative for organizations to remain vigilant and adaptive in defending against cyber threats. By understanding and countering their new adversary behaviors, we can bolster our defenses and safeguard critical assets against future intrusions.

For additional IOCs and context, including for more detail on the exploits, backdoors, and C2 involved, please see our prior post.

About the Center for Threat-Informed Defense

The Center for Threat-Informed Defense is a non-profit, privately funded research and development organization operated by MITRE Engenuity. The Center's mission is to advance the state of the art and the state of the practice in threat-informed defense globally. Comprised of participant organizations from around the globe with highly sophisticated security teams, the Center builds on MITRE ATT&CK, an important foundation for threat-informed defense used by security teams and vendors in their enterprise security operations. Because the Center operates for the public good, outputs of its research and development are available publicly and for the benefit of all.
