VMware ESXi Zero-Day Used by Chinese Espionage Actor to Perform Privileged Guest Operations on Compromised Hypervisors

cloud.google.com/blog/topics/threat-intelligence/vmware-esxi-zero-day-bypass

Mandiant

Written by: Alexander Marvi, Brad Slaybaugh, Ron Craft, Rufus Brown

Requires access to the hypervisor to exploit the vulnerability (e.g. through stolen ESXi credentials)

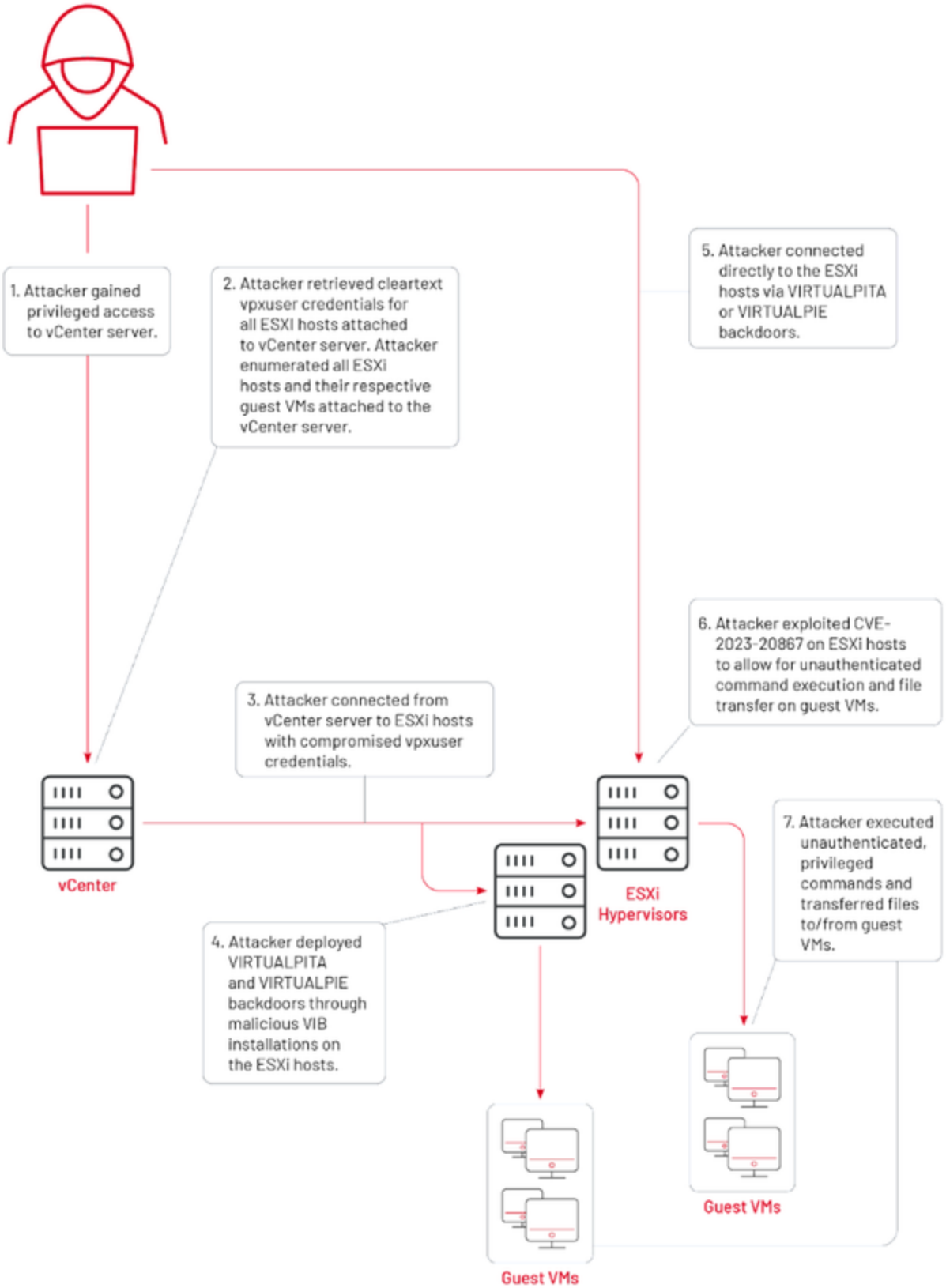
As Endpoint Detection and Response (EDR) solutions improve malware detection efficacy on Windows and Linux systems, certain state-sponsored threat actors have shifted to developing and deploying malware on systems that do not generally support EDR such as network appliances, SAN arrays, and VMware ESXi hosts.

In late 2022, Mandiant published details surrounding a novel malware system deployed by UNC3886, a Chinese cyber espionage group, which impacted VMware ESXi hosts, vCenter servers, and Windows virtual machines (VM). Through continued security research and investigations over the past year, Mandiant has discovered additional techniques UNC3886 has utilized across multiple organizations to keep out of the sights of EDR solutions including:

- Harvesting credentials for service accounts from a vCenter Server for all connected ESXi hosts from the embedded vPostgreSQL server built into vCenter Server Appliance
- Exploiting a zero-day vulnerability (CVE-2023-20867) that enabled the execution of privileged commands across Windows, Linux, and PhotonOS (vCenter) guest VMs without authentication of guest credentials from a compromised ESXi host and no default logging on guest VMs
- Deploying backdoors on ESXi hosts using an alternative socket address family, VMCI, for lateral movement and continued persistence. This address family enabled direct reconnection from any guest VM to the compromised ESXi host's backdoor regardless of network segmentation or firewall rules in place.
- Continuing to tamper with and disable logging services on impacted systems, presenting additional challenges to investigating UNC3886 in a compromised environment.

This blog post describes an expanded understanding of the attack path seen in Figure 1 and highlights the implications of both the zero-day vulnerability (CVE-2023-20867) and VMCI communication sockets the attacker leveraged to complete their goal. In a followup post, we will provide artifacts present on hosts, which indicate historical attacker activity, optional logging to track Guest Operations at the guest level, and hardening suggestions for both vCenter and ESXi solutions.

ATTACK PATH


1. Attacker gained privileged access to vCenter server.
2. Attacker retrieved cleartext vpxuser credentials for all ESXI hosts attached to vCenter server. Attacker enumerated all ESXi hosts and their respective guest VMs attached to the vCenter server.
3. Attacker connected from vCenter server to ESXi hosts with compromised vpxuser credentials.
4. Attacker deployed VIRTUALPITA and VIRTUALPIE backdoors through malicious VIB installations on the ESXi hosts.
5. Attacker connected directly to the ESXi hosts via VIRTUALPITA or VIRTUALPIE backdoors.
6. Attacker exploited CVE-2023-20867 on ESXi hosts to allow for unauthenticated command execution and file transfer on guest VMs.
7. Attacker executed unauthenticated, privileged commands and transferred files to/from guest VMs.
vCenter
ESXi Hypervisors
Guest VMs
Guest VMs
MANDIANT

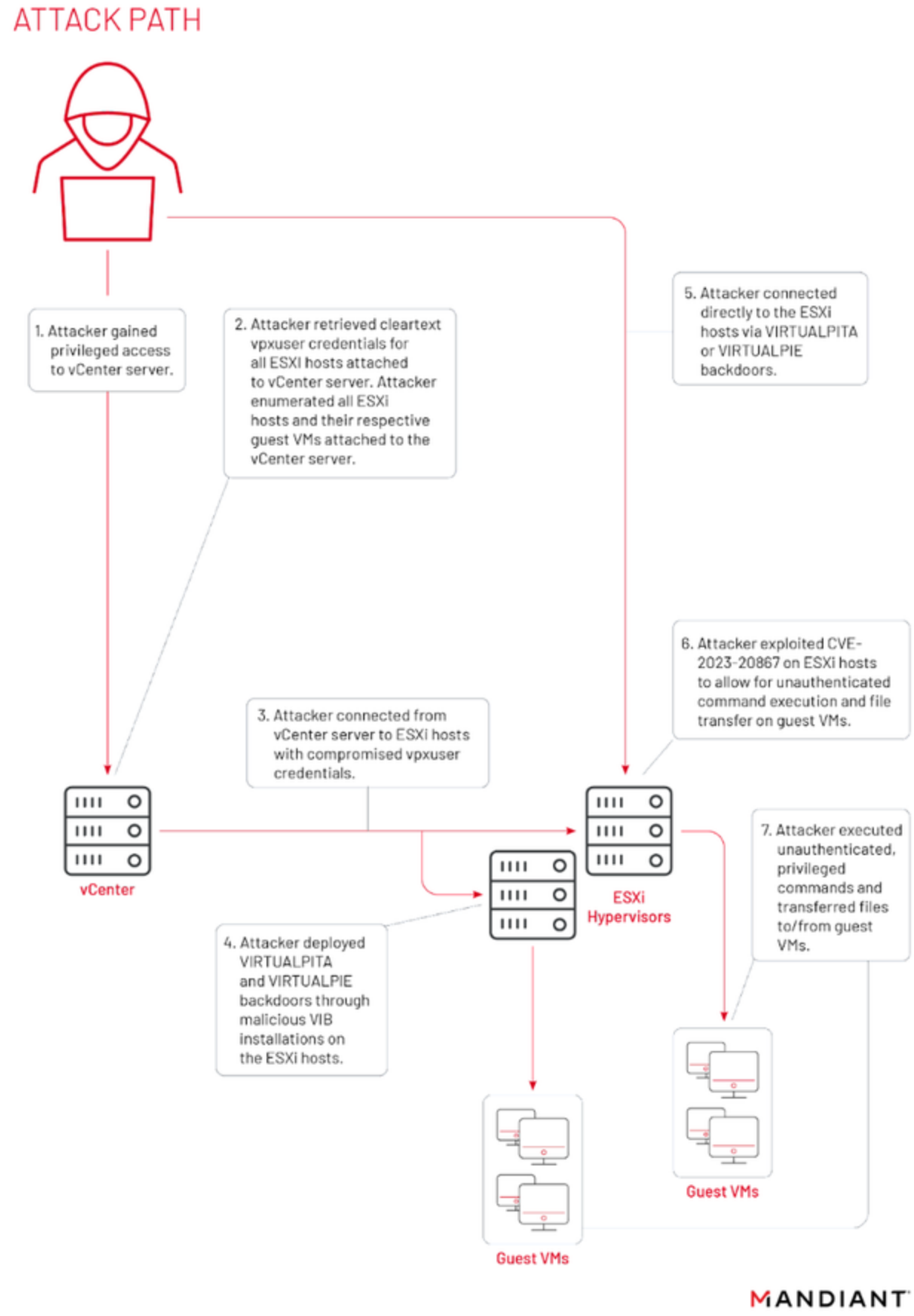


Figure 1: Expanded UNC3886 attack path

Before diving into how UNC3886 was able to harvest service account credentials for ESXi hosts on vCenter servers, the following section describes what is currently known about the threat actor associated with the attack path in Figure 1.

Attribution

UNC3886 is a highly adept Chinese cyber espionage group that has targeted and exploited zero-day vulnerabilities in firewall and virtualization technologies, which do not support EDR solutions. UNC3886 has primarily targeted defense, technology, and telecommunication organizations located in the US and APJ regions. Mandiant continues to observe UNC3886 leverage novel malware families and utilities that indicate the group has access to extensive research and support for understanding the underlying technology of appliances being targeted.

While past Mandiant blog posts related to UNC3886 have shared atomic indicators like file names and hashes, this particular attacker has been observed replacing indicators mentioned in publications in under a week after their release. For this reason, this blog post focuses on highlighting the tactics and methodologies utilized by the attacker with the aim to be able to detect and respond to this attack path regardless of the exact malware being deployed or commands being run.

The following section ties in the usage of the `vpxuser` mentioned in this blog post and describes how the attacker gained these service account credentials for ESXi machines by targeting the vCenter server the ESXi hosts were connected to.

vCenter Exploitation

In Mandiant's previous ESXi blog post, Figure 2 visualized how UNC3886 leveraged compromised ESXi hosts to execute commands on guest VMs. The attack path shows the attacker using the `vpxuser` account credentials along with the command it intended to execute on a guest VM as parameters to a Python script, `e.py`.

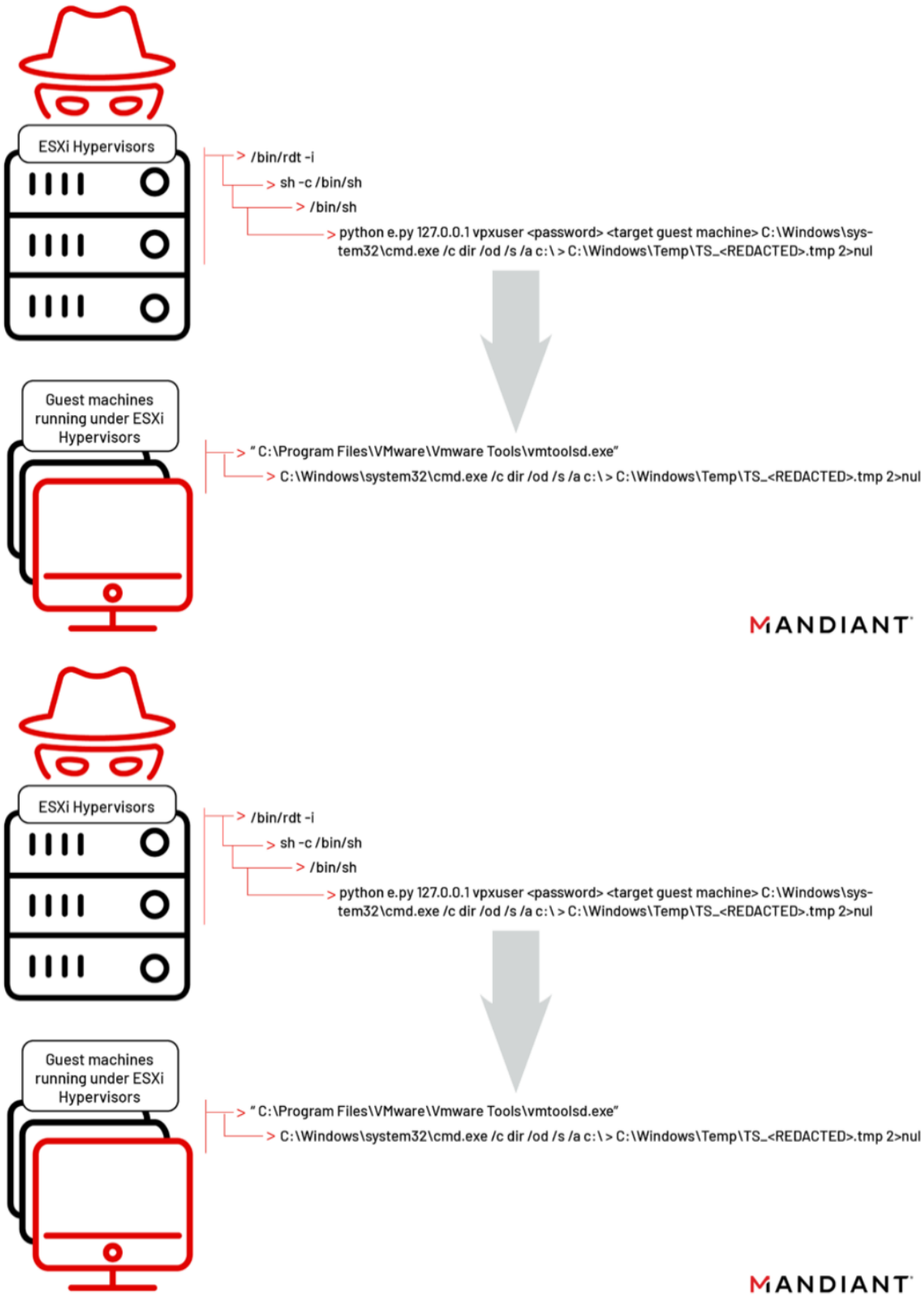


Figure 2: vpxuser account usage

Mandiant has identified additional attacker scripts that enabled UNC3886 to obtain `vpxuser` credentials, enumerate ESXi hosts and their guest VMs, and manipulate connected ESXi host firewall rules. These scripts enabled the attacker to perform the following actions:

- Obtain the cleartext passwords of all connected ESXi host's `vpxuser` service accounts from a compromised vCenter server through the connected vPostgreSQL database
- List all ESXi hosts that are attached to a vCenter server as well as the guest VMs that are hosted on each of the attached ESXi hosts
- Add or delete from the list of allowed IPs for a specified service (Default sshServer) across all connected ESXi hosts

Each of these scripts will be described in more detail in the following sections starting with how the attacker harvested the `vpxuser` credentials.

vpxuser Credential Harvesting

The `vpxuser` account is a privileged service account created on an ESXi host automatically when it is first connected to a vCenter server. The password for this user is encrypted and stored in the vPostgreSQL database on a vCenter server and by default rotates automatically every 30 days. vCenter primarily uses this service account for administrative actions such as moving VMs between different ESXi hosts and modifying the configurations of running VMs. UNC3886 utilized this service account to deploy malicious vSphere Installation Bundles (VIB) that contained either the VIRTUALPITA or VIRTUALPIE backdoor.

In March 2022, Pentera researcher Yuval Lazar published guidance related to the privilege escalation vulnerability, CVE-2022-22948, which described how the `vpxuser` credentials can be harvested on a vCenter server. While this vulnerability has been patched since the guidance was released, it is still possible to retrieve the encrypted credentials of the `vpxuser` and crack them if the attacker was able to gain root access to the vCenter server.

ESXi Host and Guest Machine Enumeration

Once the `vpxuser` credentials were acquired, the attacker used an additional script to map out the ESXi hosts and their respective guest VMs. The contents of the recovered Python script utilized the same vPostgreSQL database where the `vpxuser` credentials are stored to list all ESXi hosts that are attached to the vCenter server as well as the guest VMs that are hosted on each ESXi host.

vCenter to ESXi Host Firewall Manipulation

With the available ESXi hosts and guest VMs identified, another attacker python script was used to update the allowed list of IP addresses for any service across all ESXi hosts connected to the vCenter server this script was run against. This allowed the attacker to make quick changes to multiple ESXi hosts firewalls' at once and by default would enable IP

addresses provided to SSH directly to the ESXi hosts. It also provided the attacker with an easy way to cover their tracks by removing that same IP address once a malicious VIB was deployed, offering alternative means to reconnect later.

With the attacker's remote SSH access and the credentials for the `vpxuser` service account, the ESXi hosts were connected to and persistent backdoors were deployed. The attacker deployed these backdoors through VIB, which enabled the malware packaged inside to be executed each time the ESXi host was restarted. The VIBs contained the VIRTUALPITA and VIRTUALPIE backdoors, which listened on either TCP or VMCI ports.

The attacker scripts used to install these VIBs, perform anti-forensics to hide these installations, and exploit CVE-2023-20867 to perform unauthenticated Guest Operations are described in the following section.

ESXi Host Exploitation

Malicious VIB Installation

As additional investigations into UNC3886 progressed, Mandiant recovered installation scripts that UNC3886 used to deploy malicious VIBs to ESXi hosts. While previous blog posts Mandiant released suggested that the attacker utilized timestomping to hide their activity, as seen through specific events in the `vmkwarning.log`, the scripts discovered by Mandiant were able to confirm exactly how the attackers derived the time that was used to timestomp before installing the malicious VIBs.

The installation scripts performed the following actions on the targeted ESXi host:

1. Saved the current firewall state and system time
2. Disabled the firewall
3. Set the VIB acceptance level to PartnerSupported
4. Retrieved the VIB install time from the file: `/var/db/esximg/vibs/esx-base-*.xml`
5. Set the system clock to the retrieved VIB install time
6. Installed the malicious VIB(s) with the command: `esxcli software vib install -f --no-sig-check`
7. Re-enabled the firewall
8. Set the system clock back to the current date and time

Mandiant previously noted in 2022 that it did not identify evidence of a CVE being exploited during past investigations. As investigations into UNC3886 activity continued in 2023, Mandiant discovered that the attacker utilized a zero-day vulnerability, CVE-2023-20867, to execute commands and transfer files to and from guest VMs from a compromised ESXi host without the need for guest credentials. Additionally, the use of CVE-2023-20867 does not generate an authentication log event on the guest VM when commands are executed from

the ESXi host. Mandiant worked closely with VMware to responsibly disclose this vulnerability. For more information on this vulnerability, please see VMware's advisory, VMSA-2023-0013.

Before explaining how CVE-2023-20867 impacted ESXi hosts and connected guest machines, the concept of VMware Guest Operations, the API that this exploit impacts, is explained in further detail as follows.

VMware Guest Operations Explained

To interact with guest VMs running on compromised ESXi hosts, the attackers utilized VMware tools (vmtools) Guest Operations to interact with the VMs provisioned through the ESXi host. Vmtools Guest Operations are a feature of the vSphere API, which allow for different types of host-to-guest interactions. A list of all different Guest Operations can be seen in Table 1. (Here is a full list of GuestOperationsManager API Calls.)

Managed Object	Methods	Description
GuestAliasManager	AddGuestAlias	define alias for guest account
	ListGuestAliases	list guest aliases for specified user
	ListGuestMappedAliases	list alias map for in-guest user
	RemoveGuestAliasByCert	remove certificate associated aliases
GuestAuthManager	AcquireCredentialsInGuest	authenticate, return session object
	ReleaseCredentialsInGuest	release session object
	ValidateCredentialsInGuest	check authentication data or timeout
GuestFileManager	ChangeFileAttributesInGuest	change attributes of file in guest
	CreateTemporaryDirectoryInGuest	make a temporary directory

	CreateTemporaryFileInGuest	create a temporary file
	DeleteDirectoryInGuest	remote directory in guest OS
	DeleteFileInGuest	remove file in guest OS
	InitiateFileTransferFromGuest	start file transfer from guest OS
	InitiateFileTransferToGuest	start file transfer to guest OS
	ListFilesInGuest	list files or directories in guest
	MakeDirectoryInGuest	make a directory in guest
	MoveDirectoryInGuest	move or rename a directory in guest
	MoveFileInGuest	rename a file in guest
GuestWindowsRegistryManager	CreateRegistry KeyInGuest	create a registry key
	DeleteRegistryKeyInGuest	delete a registry key
	DeleteRegistryValueInGuest	delete a registry value
	ListReeistryKeysInGuest	list registry subkeys for a given key
	ListRegistryValuesInGuest	list registry values for a given key
	SetRegistryValueInGuest	set or create a registry value
GuestProcessManager	ListProcessesInGuest	list processes running in guest OS
	ReadEnvironmentVariableInGuest	read environment variable in guest

StartProgramInGuest	start running program in guest
TerminateProcessInGuest	stop a running process in guest

Table 1: List of vmtools Guest Operations

Normally, to run a Guest Operation, an administrator would need to authenticate with the guest machine through one of the following methods:

- NamePasswordAuthentication
- SAMLTokenAuthentication
- SSPIAuthentication
- TicketedSessionAuthentication

Mandiant noted that UNC3886 conducted Guest Operations API calls when interacting with VMs from the ESXi host. During those investigations, Mandiant did not find evidence of authentication events, such as a Windows 4624 Event ID in the Security log, when the attacker interacted with guest VMs from the ESXi host. As investigations into UNC3886 intrusions continued, Mandiant found new evidence that the attacker was able to bypass authenticating with guest VMs on the ESXi host when performing guest operations, and therefore, not generating any authentication events in available logs upon successful exploitation. This information was responsibly disclosed to VMware and, once a patch was created, was designated the label CVE-2023-20867, which is described in further detail in the following section.

CVE-2023-20867 — Unauthenticated Guest Operations from ESXi Host to Guest Machine

CVE-2023-20867 allowed the attacker to execute privileged Guest Operations on guest VMs from a compromised ESXi host without the need to authenticate with the guest VM by targeting the authentication check mechanism. Additionally, the exploit bypasses traditional logging actions performed on either the ESXi host or the guest VM. The only requirements needed to exploit this vulnerability are as follows:

- The attacker has privileged account access (such as root or the `vpxuser`) to the ESXi host
- The target guest machine has VMware Tools software installed

This exploit was able to perform successful unauthenticated Guest Operations on both Windows, Linux, and PhotonOS (vCenter) guests. The attacker accomplished the exploit by running a Python script, which injects into the running /bin/vmx process, specifically targeting

the `userCredentialType` that performs the authentication checks before executing a guest operation.

Figure 3 displays the source code of vixCommand.h from GitHub and defines 10 authentication types built into vmtools. The default challenge produced when attempting to perform a guest operation requires a username and password (VIX_USER_CREDENTIAL_NAME_PASSWORD - Type 1) before the Guest Operation can be completed. The exploit changes the `userCredentialType` challenge to Type 3, which does not perform any authentication checks before performing the Guest Operation.

```
/*
 * The types of credential we can pass with any request.
 */
#define VIX_USER_CREDENTIAL_NONE                    0
#define VIX_USER_CREDENTIAL_NAME_PASSWORD           1
#define VIX_USER_CREDENTIAL_ANONYMOUS               2
#define VIX_USER_CREDENTIAL_ROOT                    3
#define VIX_USER_CREDENTIAL_NAME_PASSWORD_OBFUSCATED  4
#define VIX_USER_CREDENTIAL_CONSOLE_USER            5
#define VIX_USER_CREDENTIAL_HOST_CONFIG_SECRET      6
#define VIX_USER_CREDENTIAL_HOST_CONFIG_HASHED_SECRET 7
#define VIX_USER_CREDENTIAL_NAMED_INTERACTIVE_USER  8
#define VIX_USER_CREDENTIAL_TICKETED_SESSION        9
#define VIX_USER_CREDENTIAL_SSPI                    10

```


```
/*
 * The types of credential we can pass with any request.
 */
#define VIX_USER_CREDENTIAL_NONE                    0
#define VIX_USER_CREDENTIAL_NAME_PASSWORD           1
#define VIX_USER_CREDENTIAL_ANONYMOUS               2
#define VIX_USER_CREDENTIAL_ROOT                    3
#define VIX_USER_CREDENTIAL_NAME_PASSWORD_OBFUSCATED  4
#define VIX_USER_CREDENTIAL_CONSOLE_USER            5
#define VIX_USER_CREDENTIAL_HOST_CONFIG_SECRET      6
#define VIX_USER_CREDENTIAL_HOST_CONFIG_HASHED_SECRET 7
#define VIX_USER_CREDENTIAL_NAMED_INTERACTIVE_USER  8
#define VIX_USER_CREDENTIAL_TICKETED_SESSION        9
#define VIX_USER_CREDENTIAL_SSPI                    10

```


Figure 3: VixCommand userCredentialTypes

Mandiant's investigation into UNC3886 uncovered additional Python scripts, which enabled them to execute actions across the guest VMs from the compromised ESXi hosts. The attacker used the Python scripts seen in Table 2 to modify the `/bin/vmx` process and perform multiple types of Guest Operations.

Filename	Description
e.py	StartProgramInGuest - Executes commands on a guest VM from an ESXi host
d.py	InitiateFileTransferFromGuest - Download files from a specified directory on a guest VM to the ESXi host
l.py lf.py	ListFilesInGuest - List all files in a specified directory on a guest VM from an ESXi host
lp.py	ListProcessesInGuest - List all running processes on a guest VM from an ESXi host
p.py	Exploits CVE-2023-20867 by modifying a single byte in the /bin/vmx process to bypass authentication on the guest VM when interacting with it from the host
pall.py	Similar functionality to p.py with extended compatibility for ESXi versions prior to 6.0
u.py	InitiateFileTransferToGuest – Uploads a file to a specified directory on a guest VM from an ESXi host

Table 2: List of attacker Python scripts utilizing VMware Tools Guest Operation API calls

As mentioned, access to an administrative account such as the vpxuser or root account of the ESXi host and vmtools installed on the guest VM are required for this exploit to succeed. The p.py script discovered during UNC3886 investigations allows the attacker to exploit CVE-2023-20867 to bypass authentication on the guest VM from the compromised ESXi host. Meanwhile, any guest VM that the attacker attempts to access through this vulnerability that doesn't have vmtools installed will result in a failed login attempt. However, once the attacker has privileged access to the ESXi host, the attacker has the capability to install vmtools on any of the guest VMs.

Currently, no logging events are generated by default when CVE-2023-20867 is successfully exploited. A successful execution of Guest Operations on a Windows guest VM does not generate a 4624 or 4634 Windows Event ID within Windows since the exploit bypasses any requirement for authentication checks on the guest VM. On Linux, the successful execution of Guest Operations using this exploit results in no actions being logged in the Linux access logs.

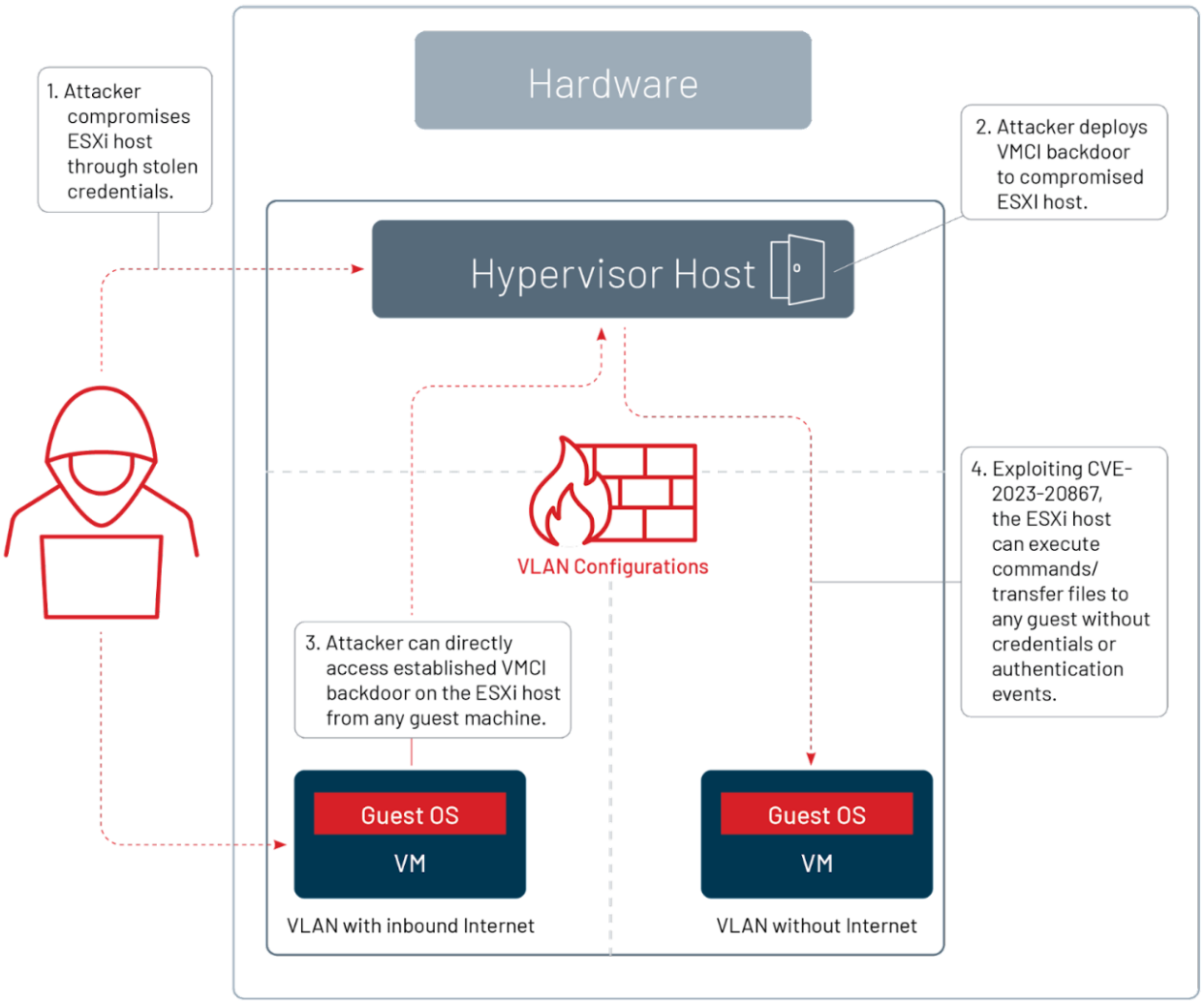
Further guidance on how to enable logging and detect both failed and successful attempts of exploitation will be found in the followup to this blog post.

VMCI Socket Abuse

What is a VMCI Socket?

Over the past year Mandiant has observed UNC3886 deploy multiple backdoors (VIRTUALGATE and VIRTUALPITA) using VMCI sockets for lateral movement and continued persistence. While this topic was covered briefly when defining VIRTUALGATE in the previous blog post, Mandiant conducted additional research to highlight what VMCI sockets are, what unique capabilities they provide the attacker, and options to potentially detect them.

At a high level, VMCI sockets are end points that enable low latency and high throughput communication between the ESXi host and its guest VMs over a channel localized on the bare metal machine running the ESXi host. Since this traffic is localized to the bare metal machine, there are no security mechanisms restricting any guest VM or ESXi host from initiating a connection with the other, as seen in Figure 4. Additionally, no traffic can be monitored outside of the guest VM's and ESXi hosts present in the virtualized environment. While the client/server communication socket has connection-oriented and connectionless variants very similar to TCP and UDP, it is invisible to commonly used networking tools such as tcpdump, netstat, nmap, and Wireshark without custom configurations as it belongs to a different socket address family.


ESXi Hypervisor VMCI Communication
Hardware
1. Attacker compromises ESXi host through stolen credentials.
2. Attacker deploys VMCI backdoor to compromised ESXI host.
Hypervisor Host
VLAN Configurations
3. Attacker can directly access established VMCI backdoor on the ESXi host from any guest machine.
4. Exploiting CVE-2023-20867, the ESXi host can execute commands/ transfer files to any guest without credentials or authentication events.
Guest OS
VM
Guest OS
VM
VLAN with inbound Internet
VLAN without Internet
MANDIANT

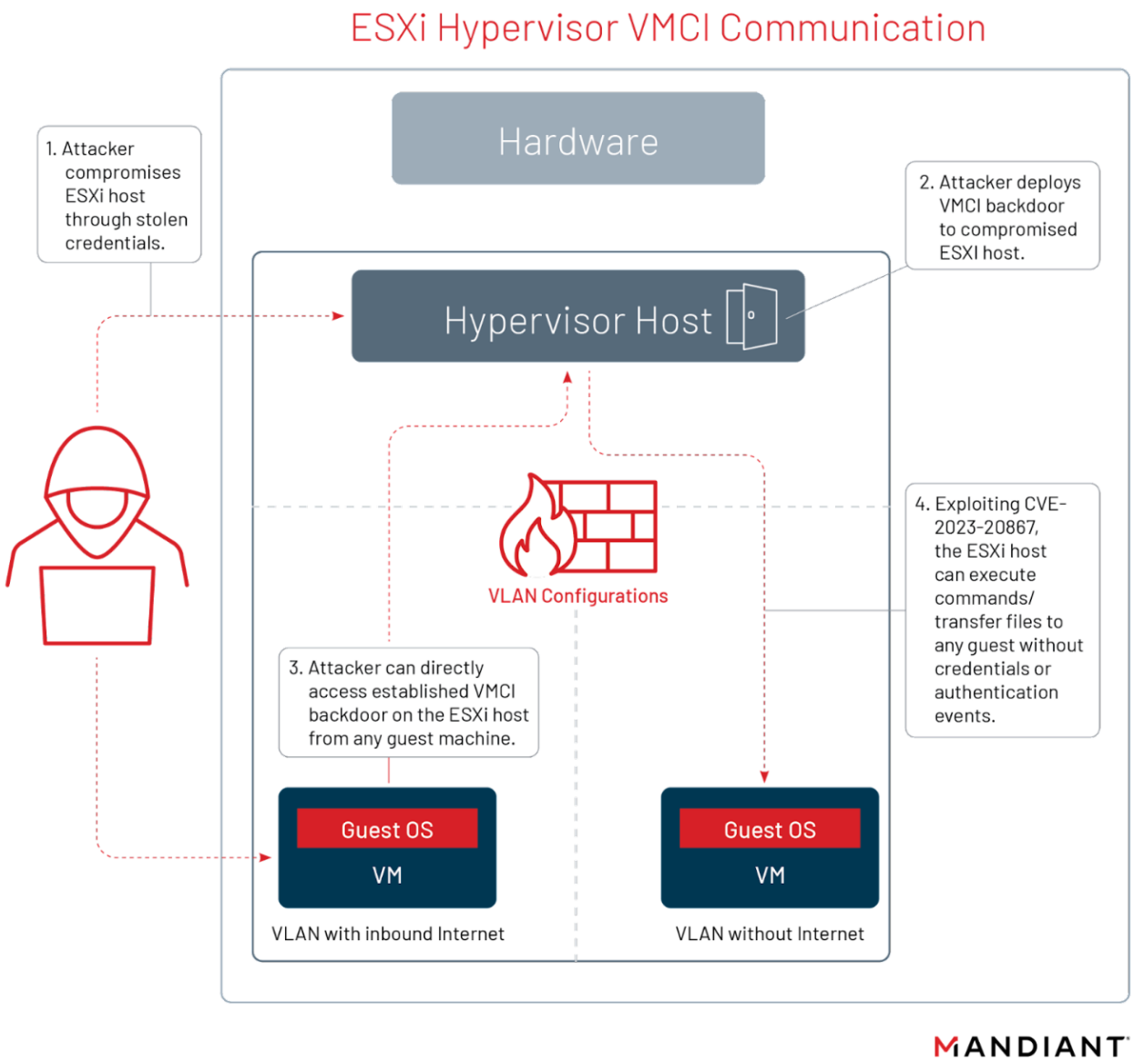


Figure 4: Overview of attacker's use of ESXi Hypervisor VMCI Communications

Before diving into VMCI routing and communications, the following section provides some background context that needs to be provided on what sockets are and how they are further categorized with address families to understand why default configurations on networking tools do not pick up VMCI activity.

Socket Address Family Definition

A socket is a two-way communication channel that acts as an endpoint to send and receive data between multiple processes within one machine or across multiple machines within a network.

Since sockets only send and receive information, standards needed to be formulated so information received could be interpreted correctly. Socket address families were created as standardized data structure formats to produce agnostic communications over data sockets.

Common address families include AF_INET and AF_UNIX, though VMCI sockets fall within their own address family. What is important to remember is that a program must have code to interpret and acknowledge information being passed to sockets, if the information does not meet the criteria of the address family being expected, it will likely be dropped/ignored.

VMCI Routable Identifiers

Similar to how private IP addresses are used to uniquely identify endpoints on a local network, VMCI sockets utilize an endpoint identifier called a Context ID (CID) to route traffic throughout the virtualized environment. The ESXi host will always have two static CIDs:

- 0 – The CID with VMCI sockets only visible to the ESXi host
- 2 – The CID that is routable to and from the guest machines when a VMCI socket is open

The CID for each guest virtual machine can be found on the ESXi host within the .vmx file for the virtual machine under `/vmfs/volume/<Volume-ID>/<Virtual-Machine-Name>.vmx`. The following lines within the .vmx file can be used to acquire the CID of the VM:

```
vmci0.id = "<CID>"
```

How Can the Attacker Utilize This?

Mandiant has confirmed UNC3886's use of multiple VMCI backdoors deployed as malicious VIBs on ESXi hosts. This open communication channel between guest and host, where either role can act as client or server, has enabled a new means of persistence to regain access on a backdoored ESXi host as long as a backdoor is deployed and the attacker gains initial access to any guest machine. This is attractive to the attacker for a multitude of reasons including but not limited to:

- The ability to bypass network segmentation normally needed to access the ESXi host
- Circumventing most security reviews for open listening ports and odd NetFlow behavior
- Regaining access to the ESXi host only requires access to any virtual machine
- In conjunction with the CVE-2023-20867, once the attacker regains access to the ESXi host they can perform unauthenticated actions with the highest privileged accounts across any virtual machine running underneath that ESXi host
- If a vCenter exists as a virtual machine underneath the ESXi host, the attacker can proceed to harvest all connected vpxuser credentials for all ESXi hosts connected to the vCenter and continue to laterally pivot across the environment.

Conclusion

As Mandiant continues to investigate UNC3886, this blog post further reinforces UNC3886's deep understanding and technical knowledge of ESXi, vCenter and VMware's virtualization platform. UNC3886 continues to target devices and platforms that traditionally lack EDR solutions and make use of zero-day exploits on those platforms. UNC3886 continues to present challenges to investigators by disabling and tampering with logging services, selectively removing log events related to their activity. The threat actors' retroactive cleanup performed within days of past public disclosures on their activity indicates how vigilant they are. Organizations must remain vigilant and ensure that they're not only monitoring their network at the operating system layer, but also continue to patch, maintain, and monitor the appliances that are running the underlying infrastructure.

Acknowledgements

Special thanks to Moritz Raabe, Joshua Kim, Matthew Maczko, Maegan Palombo, DJ Palombo, Jeremy Koppen, and Charles Carmakal for their assistance with the investigation, technical review, and creating detections for the malware families discussed in this blog post. In addition, we would also like to thank VMware for their collaboration and assistance with this research.

Posted in

Threat Intelligence