Bad VIB(E)s Part Two: Detection and Hardening within ESXi Hypervisors

mandiant.com/resources/blog/esxi-hypervisors-detection-hardening

In part one, we covered attackers' usage of malicious vSphere Installation Bundles ("VIBs") to install multiple backdoors across ESXi hypervisors, focusing on the malware present within the VIB payloads. In this installment, we will continue to elaborate further on other attacker actions such as timestomping, describe ESXi detection methodologies to dump process memory and perform YARA scans, and discuss how to further harden hypervisors to minimize the attack surface of ESXi hosts. For more details, VMware has released additional information on protecting vSphere.

ESXI Logging

Both VIRTUALPITA and VIRTUALPIE stop the `vmsyslogd` process from recording activity on startup, but multiple logging processes exist across the hypervisor which can still be used to track attacker activity.

Malicious VIB Installation

It was previously established in part one that ESXi systems do not allow for a falsified VIB file below the minimal set acceptance level, even when the acceptance level was modified in the Descriptor XML. To circumvent this, the attacker abused the `--force` flag to install malicious `CommunitySupported` VIBs. This flag adds a VIB or image profile with a lower acceptance level than required by the host.

Evidence of the `--force` flags usage to install a VIB was found across multiple locations on the ESXi hypervisor. The ESXi profile XML file records all VIBs that have been installed on the system, specifying the date, time, and flags used to install each VIB. This file is found under the path `/var/db/esximg/profile`. Figure 1contains an example of the attacker's `--force` flag usage logged in the profile XML file.

```
<imageprofile><name>(Updated) ESXi-6.7.0-20191204001-standard</name><creator>VMware, Inc.</creator>
<profileID>b066e3fa8a9711ec83592d48fd755f5d</profileID><creationtime>2022-02-10T17:34:24.587029+00:00</creationtime>

<modifiedtime>2022-02-10T17:34:24.587029+00:00</modifiedtime>
<serialno>0</serialno><description>

2022-02-10T17:34:14.680626+00:00: The following VIBs are installed:
  ata-pata-pdc20211 1.0-3vmw.670.0.0.8169922
WARNING: A --force install has been performed, the image may not be valid.

----------

2021-11-10T16:28:00.630779+00:00: The following VIBs are installed:
  lsu-lsi-lsi-mrarpid-plugin    1.0.0-9vmw.670.0.0.8169922
WARNING: A --force install has been performed, the image may not be valid.

----------

2022-02-10T21:40:26.231305+00:00: The following VIBs are installed:
  vmware-fdm    6.7.0-19300125

----------
2022-02-10T17:34:22.537277+00:00: The following VIBs are installed:
  vsanhealth    6.7.0-3.163.19184740
  esx-update    6.7.0-3.163.19195723
  esx-base  6.7.0-3.163.19195723
  vsan  6.7.0-3.163.19184739
```

Figure 1: ESXI Profile XML file with the presence of a --force installation

The log file `/var/log/ esxupdate.log` also recorded the usage of the `--force` flag when a VIB is installed. Figure 2 contains an event that logged a malicious VIB being installed with a forced installation.

```
esxupdate: root: INFO: Command = vib.install

esxupdate: root: INFO: Options = {'profile': None, 'nosigcheck': False, 'force': True, 'level': None, 'nomaintmode': False, 'downgrade':
None, 'updateonly': False, 'proxy': None, 'viburl': ['/var/tmp/ata-pata-pdc20211.vib'], 'dryrun': False, 'depot': None, 'nameid': None,
'noliveinstall': False, 'pending': None, 'oktoremove': False}

esxupdate: root: INFO: Command = vib.install

esxupdate: root: INFO: Options = {'profile': None, 'nosigcheck': False, 'force': True, 'level': None, 'nomaintmode': False, 'downgrade':
None, 'updateonly': False, 'proxy': None, 'viburl': ['/var/tmp/ata-pata-pdc20211.vib'], 'dryrun': False, 'depot': None, 'nameid': None,
'noliveinstall': False, 'pending': None, 'oktoremove': False}

esxupdate: Transaction: INFO: Final list of VIBs being installed: VMW_bootbank_ata-pata-pdc20211_1.0-3vmw.670.0.0.8169922

esxupdate: imageprofile: INFO: Adding VIB VMW_bootbank_ata-pata-pdc20211_1.0-3vmw.670.0.0.8169922 to ImageProfile (Updated) Dell ESXi-5.5-
1331820(A01)
```

Figure 2: VIB Installation with force flag in esxupdate.log

Timestomping

Mandiant observed that logs surrounding VIB installations with the `--force` flag were recorded as early as October 9, 2013, which did not align with the attack timeline. The log file `/var/log/vmkwarning.log` provided further evidence of the system time being manipulated. Figure 3 contains two (2) events that logged the system clock being modified right before and after attacker actions occurred. This behavior suggests timestomping was being performed to cover up the true time the attacker initially installed the VIBs on the machine.

```
4588-2022-03-28T11:04:26.863Z cpu27:2098674)WARNING: FTCpt: 2948: Error starting connection (8000 ms): Connection reset by peer
4589-2022-03-28T11:06:44.337Z cpu27:2098674)WARNING: FTCpt: 4975: (0 pri) Error reading socket (10/8000 ms elapsed/timeout): Connection
reset by peer
4590-2022-03-28T11:06:44.337Z cpu27:2098674)WARNING: FTCpt: 230: (0 pri) Error reading hello: Connection reset by peer
4591-2022-03-28T11:06:44.337Z cpu27:2098674)WARNING: FTCpt: 2689: (0 pri) Error reading hello: Connection reset by peer
4592-2022-03-28T11:06:44.337Z cpu27:2098674)WARNING: FTCpt: 2948: Error starting connection (8000 ms): Connection reset by peer
4593:2022-04-11T05:14:30.473Z cpu31:2099712 opID=15f171e2)WARNING: NTPClock: 1251: system clock stepped to 1548235410.000000000, but delta
101418659 > 172800 seconds
4594-2019-01-23T09:23:44.287Z cpu20:4553298)ALERT: Attempting to install an image profile with validation disabled. This may result in an
image with unsatisfied dependencies, file or package conflicts, and potential security violations.
4595-2019-01-23T09:23:44.287Z cpu20:4553298)ALERT: Attempting to install an image profile bypassing signing and acceptance level
verification. This may pose a large security risk.
4596-2019-01-23T09:23:54.065Z cpu22:4553448)ALERT: Attempting to mount a tardisk from a vib without valid signature, this may result in
security breach.
4597-2019-01-23T09:23:54.104Z cpu10:4553453)WARNING: VisorFS: 1093: Attempt to remove non sticky dir/file from tar mount
4598-2019-01-23T09:24:06.626Z cpu22:4553465)WARNING: VisorFS: 1093: Attempt to remove non sticky dir/file from tar mount
--
5021-2019-01-23T09:24:07.018Z cpu20:4553298)WARNING: VisorFS: 1093: Attempt to remove non sticky dir/file from tar mount
5022-2019-01-23T09:24:07.018Z cpu20:4553298)WARNING: VisorFS: 1093: Attempt to remove non sticky dir/file from tar mount
5023-2019-01-23T09:24:07.018Z cpu20:4553298)WARNING: VisorFS: 1093: Attempt to remove non sticky dir/file from tar mount
5024-2019-01-23T09:24:07.018Z cpu20:4553298)WARNING: VisorFS: 1093: Attempt to remove non sticky dir/file from tar mount
5025-2019-01-23T09:24:07.018Z cpu20:4553298)WARNING: VisorFS: 1093: Attempt to remove non sticky dir/file from tar mount
5026:2019-01-23T09:24:33.065Z cpu26:2347016 opID=1ecc0b6c)WARNING: NTPClock: 1251: system clock stepped to 1649654133.000000000, but delta
101418661 > 172800 seconds
5027:2022-04-11T05:20:41.503Z cpu22:2099712 opID=3b1f0a96)WARNING: NTPClock: 1251: system clock stepped to 1623179981.000000000, but delta
26474460 > 172800 seconds
```

Figure 3: vmkwarning.log recording system time modification

Creation of sysclog

Analyzing the VIRTUALPITA sample `rhttpproxy-io` (2c28ec2d541f555b2838099ca849f965), it was found that the sample listened over the VMCI port number 18098. Once the listener is setup, the malware fetches the system's CID (context ID) by issuing IOCTL request code 1977. The PID of the backdoor, CID and listening port are then logged to `/var/log/sysclog` in the following format `[<date/timestamp>]\n\r[!]<<PID>>:<CID>:<port>\n\n` as seen in Figure 4.

```
[2021-06-08 19:20:05] [!]<4553865>:2:18098
```

Figure 4: Sample of sysclog

Guest Machine Interaction

Further interactions between hypervisors and their respective guest machines were discovered within multiple logs named `vmware.log`. These logs, located at the following path `/vmfs/volumes/…/<virtual machine hostname>/vmware.log`, record basic operations between the host and hypervisor that were not logged on the endpoint. Actions recorded by this log include guest machine logins, file/directory creation and deletion,

command execution, and file transfer between guest machine and hypervisor. To focus on interactions between the hypervisor and its guest machines in the `vmware.log`, filter for lines containing GuestOps.

VIB Verification at Scale

The previous blog post touched on using the command `esxcli software vib signature verify` to identify any VIBs that do not pass the signature verification check made by the ESXi hypervisor. Alternative VIB configurations exist that would be able to circumvent the signature verification check. Mandiant confirmed that when a VIB is installed as `CommunitySupported`, the `Signature Verification` field will label it as `Succeeded` if the payload is not tampered with after installation. This means a VIB could be created without any validation from VMWare or its partners and still be labelled as validated.

To account for properly signed `CommunitySupported` VIBs and other anomalous configurations which could indicate malicious activity, all VIBs in the environment can be compared with a list of known good VIBs. A matrix created by *VMware Front Experience* breaks down the names and builds of each VIB expected to be present by default in the respective ESXi build. Each time a VIB is changed across ESXi builds the matrix links to the official VMware patch release notes which state the adding, modification, or removal of that VIB. A sample of this matrix can be seen in Figure 5.

Release Date (Info)	2022-07-12	2022-06-14		2022-01-25
Imageprofile	ESXi-6.7.0-20220704001-standard	ESXi-6.7.0-20220604001-standard	ESXi-6.7.0-20220601001s-standard	ESXi-6.7.0-20220104001-standard
Build No.	19997733 (all)	19898906 (all)	19898894 (security only)	19195723 (all)
ata-libata-92	3.00.9.2-16vmw.670.0.0.8169922	3.00.9.2-16vmw.670.0.0.8169922	3.00.9.2-16vmw.670.0.0.8169922	3.00.9.2-16vmw.670.0.0.8169922
ata-pata-amd	0.3.10-3vmw.670.0.0.8169922	0.3.10-3vmw.670.0.0.8169922	0.3.10-3vmw.670.0.0.8169922	0.3.10-3vmw.670.0.0.8169922
ata-pata-atiixp	0.4.6-4vmw.670.0.0.8169922	0.4.6-4vmw.670.0.0.8169922	0.4.6-4vmw.670.0.0.8169922	0.4.6-4vmw.670.0.0.8169922
ata-pata-cmd64x	0.2.5-3vmw.670.0.0.8169922	0.2.5-3vmw.670.0.0.8169922	0.2.5-3vmw.670.0.0.8169922	0.2.5-3vmw.670.0.0.8169922
ata-pata-hpt3x2n	0.3.4-3vmw.670.0.0.8169922	0.3.4-3vmw.670.0.0.8169922	0.3.4-3vmw.670.0.0.8169922	0.3.4-3vmw.670.0.0.8169922
ata-pata-pdc2027x	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922
ata-pata-serverworks	0.4.3-3vmw.670.0.0.8169922	0.4.3-3vmw.670.0.0.8169922	0.4.3-3vmw.670.0.0.8169922	0.4.3-3vmw.670.0.0.8169922
ata-pata-sil680	0.4.8-3vmw.670.0.0.8169922	0.4.8-3vmw.670.0.0.8169922	0.4.8-3vmw.670.0.0.8169922	0.4.8-3vmw.670.0.0.8169922
ata-pata-via	0.3.3-2vmw.670.0.0.8169922	0.3.3-2vmw.670.0.0.8169922	0.3.3-2vmw.670.0.0.8169922	0.3.3-2vmw.670.0.0.8169922
block-cciss	3.6.14-10vmw.670.0.0.8169922	3.6.14-10vmw.670.0.0.8169922	3.6.14-10vmw.670.0.0.8169922	3.6.14-10vmw.670.0.0.8169922
bnxtnet	20.6.101.7-24vmw.670.3.73.14320388	20.6.101.7-24vmw.670.3.73.14320388	20.6.101.7-24vmw.670.3.73.14320388	20.6.101.7-24vmw.670.3.73.14320388
bnxtroce	20.6.101.0-20vmw.670.1.28.10302608	20.6.101.0-20vmw.670.1.28.10302608	20.6.101.0-20vmw.670.1.28.10302608	20.6.101.0-20vmw.670.1.28.10302608
brcmfcoe	11.4.1078.26-14vmw.670.3.159.18828794	11.4.1078.26-14vmw.670.3.159.18828794	11.4.1078.26-14vmw.670.3.159.18828794	11.4.1078.26-14vmw.670.3.159.18828794
char-random	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922
cpu-microcode	6.7.0-3.170.19898894	6.7.0-3.170.19898894	6.7.0-3.170.19898894	6.7.0-3.155.18812553
ehci-ehci-hcd	1.0-4vmw.670.0.0.8169922	1.0-4vmw.670.0.0.8169922	1.0-4vmw.670.0.0.8169922	1.0-4vmw.670.0.0.8169922
elx-esx-libelxima.so	11.4.1184.2-3.89.15160138	11.4.1184.2-3.89.15160138	11.4.1184.2-3.89.15160138	11.4.1184.2-3.89.15160138
elxiscsi	11.4.1174.0-2vmw.670.0.0.8169922	11.4.1174.0-2vmw.670.0.0.8169922	11.4.1174.0-2vmw.670.0.0.8169922	11.4.1174.0-2vmw.670.0.0.8169922
elxnet	11.4.1097.0-5vmw.670.3.73.14320388	11.4.1097.0-5vmw.670.3.73.14320388	11.4.1097.0-5vmw.670.3.73.14320388	11.4.1097.0-5vmw.670.3.73.14320388
esx-base	6.7.0-3.178.19997733	6.7.0-3.174.19898906	6.7.0-3.170.19898894	6.7.0-3.163.19195723
esx-dvfilter-generic-fastpath	6.7.0-0.0.8169922	6.7.0-0.0.8169922	6.7.0-0.0.8169922	6.7.0-0.0.8169922
esx-ui	1.33.7-15803439	1.33.7-15803439	1.33.7-15803439	1.33.7-15803439
esx-update	6.7.0-3.178.19997733	6.7.0-3.174.19898906	6.7.0-3.170.19898894	6.7.0-3.163.19195723
esx-xserver	6.7.0-3.174.19898906	6.7.0-3.174.19898906	6.7.0-3.73.14320388	6.7.0-3.73.14320388
hid-hid	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922	1.0-3vmw.670.0.0.8169922
i40en	1.8.1.9-2vmw.670.3.73.14320388	1.8.1.9-2vmw.670.3.73.14320388	1.8.1.9-2vmw.670.3.73.14320388	1.8.1.9-2vmw.670.3.73.14320388
iavmd	1.2.0.1011-2vmw.670.0.0.8169922	1.2.0.1011-2vmw.670.0.0.8169922	1.2.0.1011-2vmw.670.0.0.8169922	1.2.0.1011-2vmw.670.0.0.8169922
igbn	0.1.1.0-5vmw.670.3.73.14320388	0.1.1.0-5vmw.670.3.73.14320388	0.1.1.0-5vmw.670.3.73.14320388	0.1.1.0-5vmw.670.3.73.14320388

Figure 5: Sample of Known Good VIB Matrix

ESXI Detection Methodologies

While ESXi shares many similarities to Linux (commands, directory structure, etc.), it is entirely its own operating system known as VMkernel, meaning popular methods to scan the filesystem and dump process memory do not work. Mandiant has formulated alternative detections methods to attempt to provide investigators with better visibility into ESXi hypervisors during future incidents.

Remote ESXi YARA Scanning with SSHFS

Multiple YARA rules were generated for the detection of VIRTUALPITA and VIRTUALPIE across Linux and ESXi environments and can be found in the first part of this blog post. These detections have two caveats to them based on the storage and execution of the malware. If the attacker is launching either malware family from a VIB on ESXi, the sample within the VIB will not be detected due to being compressed in the .vgz format. Secondly, if the binary is running in memory but deleted from disk, the binary will not be detected by YARA's file system scan.

Since YARA does not run directly on ESXi hosts, Mandiant utilized `sshfs` to perform remote YARA scanning of ESXi hypervisors.

Prerequisites

Note: All behaviors of ESXi and the methodology to dump memory have been confirmed for ESXi 6.7, no other versions at this time have been tested.

Before scanning the ESXi machine a few prerequisites must be met. For the ESXi machine which the memory is being dumped, you must have both:

- Root Access to the machine
- SSH Enabled on the ESXi Server

Once the ESXi machine is correctly configured, a Linux machine must be setup to be able to communicate over SSH with the ESXi machine. This Linux machine must also install:

- sshfs
- yara

Performing the YARA Scan

Note: Since YARA will naturally recursively scan directories and sshfs pulls files back as they are accessed, scanning the entire ESXi file system can take a long time depending on network bandwidth. This method of scanning is only suggested if a strong and stable network connection is present.

Linux Commands

Description	Commands
Create a directory to mount the ESXi machine on	`> mkdir /mnt/yara`
Mount the ESXi root directory to the Linux machine mount point using sshfs	`> sshfs -o allow_other,default_permissions root@<Insert ESXi IP Address>:/ /mnt/yara`
Scan the mount point which the ESXi system is attached to	`> yara -r <Provided YARA Rule> <scope of scan>`

Dumping ESXi Process Memory

When attempting to dump the process memory from a ESXi hypervisor like you would a Linux machine, it will quickly become apparent that the /proc/ directory will be either empty or containing a single PID of the commands used to attempt to dump the memory. To recover process memory from ESXi (and potentially the full binary itself), a mixture of the native tool gdbserver and a github tool called core2ELF64 can be utilized.

Prerequisites

Note: All behaviors of ESXi and the methodology to dump memory have been confirmed for ESXi 6.7, no other versions at this time have been tested.

Before dumping the process memory a few prerequisites must be met. For the ESXi machine which, you must have both:

- Root Access to the machine
- SSH Enabled on the ESXi Server

Once the ESXi machine is correctly configured, a Linux machine must be setup to be able to communicate over SSH with the ESXi machine. This Linux machine must also install:

- gdb
- core2ELF64

Dumping Memory

Note: The ports to listen and port forward through are arbitrary (Rule of Thumb: Keep between 1024-25565 to avoid commonly used ports), for this walkthrough the listening port will be 6000 and the forwarding port will be 7000.

To dump the ESXi process memory, gdbserver will be utilized to hook into the currently running process, specified by PID, and listen on an arbitrary port.

ESXi Commands

Description	Commands
A preemptive check used to make sure that the PID you will be collecting in the next command is the intended one. Please make sure that the output of this statement only shows the process you intend to dump the memory for.	`> ps -Tcistv | grep -e "<Binary to Dump>"`
Attaches gdbserver to the PID specified in the list processes command, listening on port 6000 for gdb to connect to.	`> gdbserver –attach 127.0.0.1:6000 `ps -Tcjstv | grep -e "<Binary to Dump>" | awk '{print $1}' | head -n 1``

Once listening, the Linux machine will create an SSH tunnel (Port Forward) to the listening port on the ESXi server, where gdb will be used to create a core dump of the process specified.

Linux Commands

Description	Commands
Set up an SSH tunnel from the Linux machine to the listening port of the ESXi Server gdbserver process.	`> ssh -L 1336:127.0.0.1:6000 -f -N <acct on ESX>@<IP of ESX>`

Description	Commands
Launch gdb	`> gdb`
Within the gdb shell, connect to the gdbserver instance. If at any point you have successfully ran this command and leave the gdb shell, you will need to exit and relaunch the gdbserver process on ESXi to reconnect.	`(gdb) > target remote localhost:1336`
Create a core dump file of the attach processes' memory in the working directory. The output file should be the following syntax "core.[0-9]{7}".	`?? () > gcore`

Process Extraction

Once a core file is created, the Github project core2ELF64 can be used to reconstruct the program.

Linux Commands

Description	Commands
Set up an SSH tunnel from the Linux machine to the listening port of the ESXi Server gdbserver process. In the event of the program not being able to recover the first segment, choose the next available segment possible (Smallest Number)	`> core2ELF64 <core file> <Desired Output Name>`

Sources

Hooking into ESXi processes with gdbserver

Hardening ESXi

Network Isolation

When configuring networking on the ESXi hosts, only enable VMkernel network adapters on the isolated management network. VMkernel network adapters provide network connectivity for the ESXi hosts and handle necessary system traffic for functionality such as vSphere vMotion, vSAN and vSphere replication. Ensure that all dependent technologies such as

vSANs and backup systems that the virtualization infrastructure will use are available on this isolated network. If possible, use dedicated management systems exclusively connected to this isolated network to conduct all management tasks of the virtualization infrastructure.

Identity and Access Management

Consider decoupling ESXi and vCenter Servers from Active Directory and use vCenter Single Sign-On. Removing ESXi and vCenter from Active Directory will prevent any compromised Active Directory accounts from being able to be used to authenticate directly to the virtualization infrastructure. Ensure administrators use separate and dedicated accounts for managing and accessing the virtualized infrastructure. Enforce multi-factor authentication (MFA) for all management access to vCenter Server instances and store all administrative credentials in a Privileged Access Management (PAM) system.

Services Management

To further restrict services and management of ESXi hosts, implement lockdown mode. This ensures that ESXi hosts can only be accessed through a vCenter Server, disables some services, and restricts some services to certain defined users. Configure the built-in ESXi host firewall to restrict management access only from specific IP addresses or subnets that correlate to management systems on the isolated network. Determine the appropriate risk acceptance level for vSphere Installable Bundles (VIBs) and enforce acceptance levels in the Security Profiles for ESXi hosts. This protects the integrity of the hosts and ensures unsigned VIBs cannot be installed.

Log Management

Centralized logging of ESXi environments is critical, both to proactively detect potential malicious behavior and investigate an actual incident. Ensure all ESXi host and vCenter Server logs are being forwarded to the organization’s SIEM solution. This provides visibility into security events beyond that of normal administrative activity.

Acknowledgements

Special thanks to Brad Slaybaugh, Joshua Kim, Zachary Smith, Jeremy Koppen, and Charles Carmakal for their assistance with the investigation, technical review, and creating detections/investigative methodologies for the malware families discussed in this blog post. In addition, we would also like to thank VMware their collaboration on this research.