Even when we assume that a cloud service provider is trustworthy, many users either ignore or underestimate

the danger posed by other sources of concern. One of them, especially critical to the IaaS

cloud delivery model, is image sharing. For example, a user of AWS has the option to choose between

Amazon Machine Images (AMIs), accessible through the Quick Start or the Community AMI menus of

the EC2 service. The option of using one of these AMIs is especially tempting for a first-time or less

sophisticated user.

First, let’s review the process to create an AMI. We can start from a running system, from another

AMI, or from the image of a VM and copy the contents of the file system to the S3, the so-called

bundling. The first of the three steps in bundling is to create an image, the second step is to compress

and encrypt the image, and the last step is to split the image into several segments and then upload the

segments to the S3.

Two procedures for the creation of an image are available: ec2-bundle-image and ec2-

bundle-volume. The first is used for images prepared as loopback files10 when the data is transferred

to the image in blocks. To bundle a running system, the creator of the image can use the second procedure

when bundling works at the level of the file system and files are copied recursively to the image.

To use an image, a user has to specify the resources, provide the credentials for login, provide a

firewall configuration, and specify the region, as discussed in Section 3.1. Once instantiated, the user

is informed about the public DNS and the virtual machine is made available. A Linux system can be

accessed using ssh at port 22, whereas the Remote Desktop at port 3389 is used for Windows.

A recent paper reports on the results of an analysis carried out over a period of several months, from

November 2010 to May 2011, of over 5,000 AMIs available through the public catalog at Amazon [38].

Many of the analyzed images allowed a user to undelete files and recover credentials, private keys, or

other types of sensitive information with little effort and using standard tools. The results of this study

were shared with Amazon’s Security Team, which acted promptly to reduce the threats posed to AWS

users.

The details of the testing methodology can be found in [38]. Here we only discuss the results. The

study was able to audit some 5,303 images out of the 8,448 Linux AMIs and 1,202 Windows AMIs at

Amazon sites in the United States, Europe, and Asia. The audit covered software vulnerabilities and

security and privacy risks.

The average duration of an audit was 77 minutes for a Windows image and 21 minutes for a Linux

image; the average disk space used was about 1 GB and 2.7 GB, respectively. The entire file system of a

Windows AMI was audited because most malware targetsWindows systems. Only directories containing

executables for Linux AMIs were scanned; this strategy and the considerably longer start-up time of

Windows explain the time discrepancy of the audits across the types of AMIs.

The software vulnerability audit revealed that 98% of the Windows AMIs (249 out of 253) and

58% of Linux AMIs (2,005 out of 3,432) audited had critical vulnerabilities. The average number

of vulnerabilities per AMI were 46 for Windows and 11 for Linux. Some of the images were

rather old; 145, 38, and 2 Windows AMIs and 1,197, 364, and 106 Linux were older than two,

three, and four years, respectively. The tool used to detect vulnerabilities, Nessus, available from

www.tenable.com/productus/nessus, classifies the vulnerabilities based on their severity

in four groups, at levels 0–3. The audit reported only vulnerabilities of the highest severity level, e.g.,

remote code execution.

Three types of security risks were analyzed: (1) backdoors and leftover credentials, (2) unsolicited

connections, and (3) malware. An astounding finding is that about 22% of the scanned Linux AMIs

contained credentials allowing an intruder to remotely log into the system. Some 100 passwords, 995

ssh keys, and 90 cases in which both passwords and keys could be retrieved were identified.

To rent a Linux AMI, a user must provide the public part of the ssh key, and this key is stored in

the authorized\_keys in the home directory. This opens a backdoor for a malicious creator of an

AMI who does not remove his own public key from the image and can remotely log into any instance

of this AMI. Another backdoor is opened when the ssh server allows password-based authentication

and the malicious creator of an AMI does not remove his own password. This backdoor is opened even

wider as one can extract the password hashes and then crack the passwords using a tool such as John

the Ripper (see www.openwall.com/john).

Another threat is posed by the omission of the cloud-init script that should be invoked when

the image is booted. This script, provided by Amazon, regenerates the host key an ssh server uses to

identify itself; the public part of this key is used to authenticate the server. When this key is shared

among several systems, these systems become vulnerable to man-in-the middle11 attacks. When this script does not run, an attacker can use the NMap tool12 to match the ssh keys discovered in the AMI

images with the keys obtained via NMap. The study reports that the authors were able to identify more

than 2,100 instances following this procedure.

Unsolicited connections pose a serious threat to a system. Outgoing connections allow an outside

entity to receive privileged information, e.g., the IP address of an instance and events recorded by a

syslog daemon to files in the var/log directory of a Linux system. Such information is available only

to users with administrative privileges. The audit detected two Linux instances with modified syslog

daemons, which forwarded to an outside agent information about events such as login and incoming

requests to aWeb server. Some of the unsolicited connections are legitimate – for example, connections

to a software update site. It is next to impossible to distinguish legitimate from malicious connections.

Malware, including viruses, worms, spyware, and trojans, were identified using ClamAV, a software

tool with a database of some 850,000 malware signatures, available from www.clamav.net. Two

infectedWindows AMIs were discovered, one with a Trojan-Spy (variant 50112) and a second one with a

Trojan-Agent (variant 173287). The first trojan carries out keylogging and allows stealing data from the

files system and monitoring processes; the AMI also included a tool called Trojan.Firepass to decrypt

and recover passwords stored by the Firefox browser.

The creator of a shared AMI assumes some privacy risks; his private keys, IP addresses, browser

history, shell history, and deleted files can be recovered from the published images. A malicious agent

can recover the AWS API keys that are not password protected. Then the malicious agent can start

AMIs and run cloud applications at no cost to herself, since the computing charges are passed on to

the owner of the API key. The search can target files with names such as pk − [0 − 9A − Z]∗

.pem or

cert − [0 − 9A − Z]∗

.pem used to store API keys.

Another avenue for a malicious agent is to recover ssh keys stored in files named id\_dsa and id\_rsa.

Though ssh keys can be protected by a passphrase,13 the audit determined that the majority of ssh

keys (54 out of 56) were not password protected.

Recovery of IP addresses of other systems owned by the same user requires access to the lastlog

or the lastb databases. The audit found 187 AMIs with a total of more than 66,000 entries in their

lastb databases. Nine AMIs contained Firefox browser history and allowed the auditor to identify the

domains contacted by the user.

In addition, 612 AMIs contained at least one shell history file. The audit analyzed 869 history files

named ∼/.history, ∼/.bash\_history, and ∼/.sh\_history, containing some, 160,000 lines of command

history, and identified 74 identification credentials. Users should be aware that when HTTP is used to

transfer information from a user to aWeb site, the GET requests are stored in the logs of theWeb server.

Passwords and credit card numbers communicated via a GET request can be exploited by a malicious

agent with access to such logs. When remote credentials such as the DNS management password are

available, a malicious agent can redirect traffic from its original destination to her own system.

Recovery of deleted files containing sensitive information poses another risk for the provider of an

image. When the sectors on the disk containing sensitive information are actually overwritten by another

file, recovery of sensitive information is much harder. To be safe, the creator of the image effort should

use utilities such as shred,scrub,zerofree, or wipe to make recovery of sensitive information

next to impossible. If the image is created with the block-level tool discussed at the beginning of

this section, the image will contain blocks of the file system marked as free; such blocks may contain

information from deleted files.The audit process was able to recover files from98%of theAMIs using the

exundelete utility. The number of files recovered from an AMI was as low as 6 and as high as 40,000.

We conclude that the users of published AMIs as well as the providers of images may be vulnerable

to a wide range of security risks and must be fully aware of the dangers posed by image sharing.