TREND MICRO

Securing Your Journey
to the Cloud

SatanCloud
A Journey into the Privacy and Security Risks of a Cloud Computing

Marco Balduzzi, MSc./Ph.D. • Senior Threat Researcher

TREND MICRO

Who am I?

- From Bergamo (Italy)
  - MSc. in Computer Engineering
- Télécom ParisTech (France)
  - Ph.D. in Applied System Security
- 10+ years experience in IT Security
- Engineer and consultant for different international firms
  - Senior Threat Researcher @ TrendMicro
- Co-founder of BGLug, Applied UniLab, (ex) SPINE Group, free software developer, hacking groups

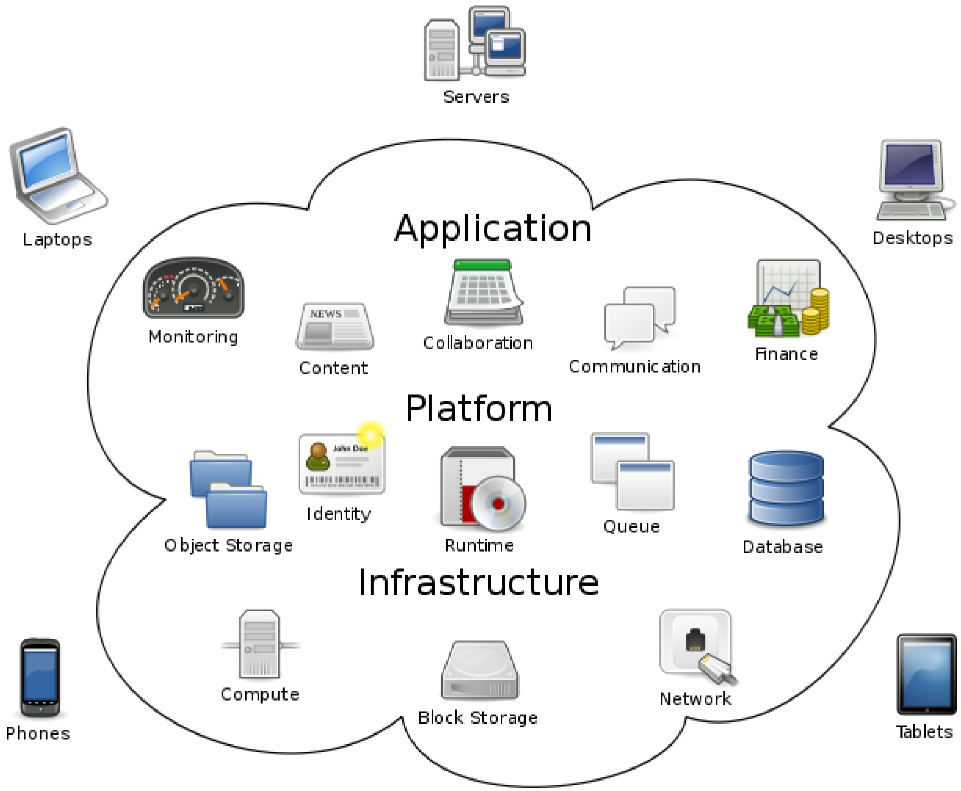
[http://www.iseclab.org/people/**embyte**](http://www.iseclab.org/people/embyte)

Roadmap

- Introduction
  - Cloud Computing
  - IaaS and Amazon EC2
- Security Problem definition
- *SatanCloud*
  - Automated analysis & testing
- Experiments
  - Findings
- Lessons learned
- Conclusions

What is Cloud Computing?

- *The delivery of* ***computing as a service rather than a product****, whereby shared resources, software, and information are provided to computers and other devices as a* ***utility over a network*** *(Internet).* *[wikipedia]*



Cloud, an old new concept

- Parallel, distributed and grid computing have been around for a while
  - Scientists, governments, international organizations, military
  - Urban planning, weather forecasts, economic modeling, etc…
- Now, cloud computing is a commodity
  - Who does not use the cloud nowadays?
- Ready-to-go services

3 Models of Cloud Services

- Software as a Service (**SaaS**): software
  - e.g. CRM, email, games, virtual desktops
    - Google Apps, Salesforce CRM, Dropbox
- Platform as a Service (**PaaS**): computing or solution platform
  - e.g. programming language execution environments, databases, web servers
    - Microsoft's Azure, Google's AppEngine.
- Infrastructure as a Service (**IaaS**): computers (physical/virtual), storage, firewalls or networks
    - Amazon EC2, Rackspace Cloud, Joyent Smart Machines

Infrastructure as a Service

- **Remote access to virtualized server** images on an hourly/monthly basis
- Amazon's Elastic Compute Cloud (EC2)
- Competitors (Jason Read @ CloudHarmony.com)
  - Storm on Demand: $100/mo
  - Voxel VoxCLOUD: $144/mo
  - Linode VPS: $160/mo
  - ThePlanet Cloud Servers: $169/mo
  - Zerigo: $173/mo
  - Rackspace Cloud: $175/mo
  - NewServers Bare Metal Cloud: $180/mo
  - SoftLayer CloudLayer Computing: $199/mo
  - Terremark vCloud Express: $202/mo
  - ReliaCloud: $230/mo
  - GoGrid: $232/mo
  - Joyent Smart Machines: $500/mo

Amazon EC2 [1/3]

- Infrastructure-as-a-Service platform
- Users can **rent** **A**mazon **M**achine **I**mages (called **AMIs**) on an hourly basis
  - Provided an online catalog
  - Web interface and APIs
- Users can **publish** AMIs to the Cloud
  - **1**. Amazon itself
  - **2**. individuals
  - **3**. third-party companies (can charge extra costs via *Amazon DevPay)*

Amazon EC2 [2/3]

- AMI can be built from…
  - … a live system
  - … a virtual machine image (ISO)
  - … or another AMI (by copying the file system contents to S3)
- To start an Image, the user configures:
  - Credentials
  - Resources: processing, memory, IO performance
  - Region: US East, US West, Europe, Singapore, Tokyo
  - Inbound firewall
- Three pricing models
  - Fixed pricing
  - Subscription
  - Spot instances (price changes according to load)

Amazon EC2 [3/3]

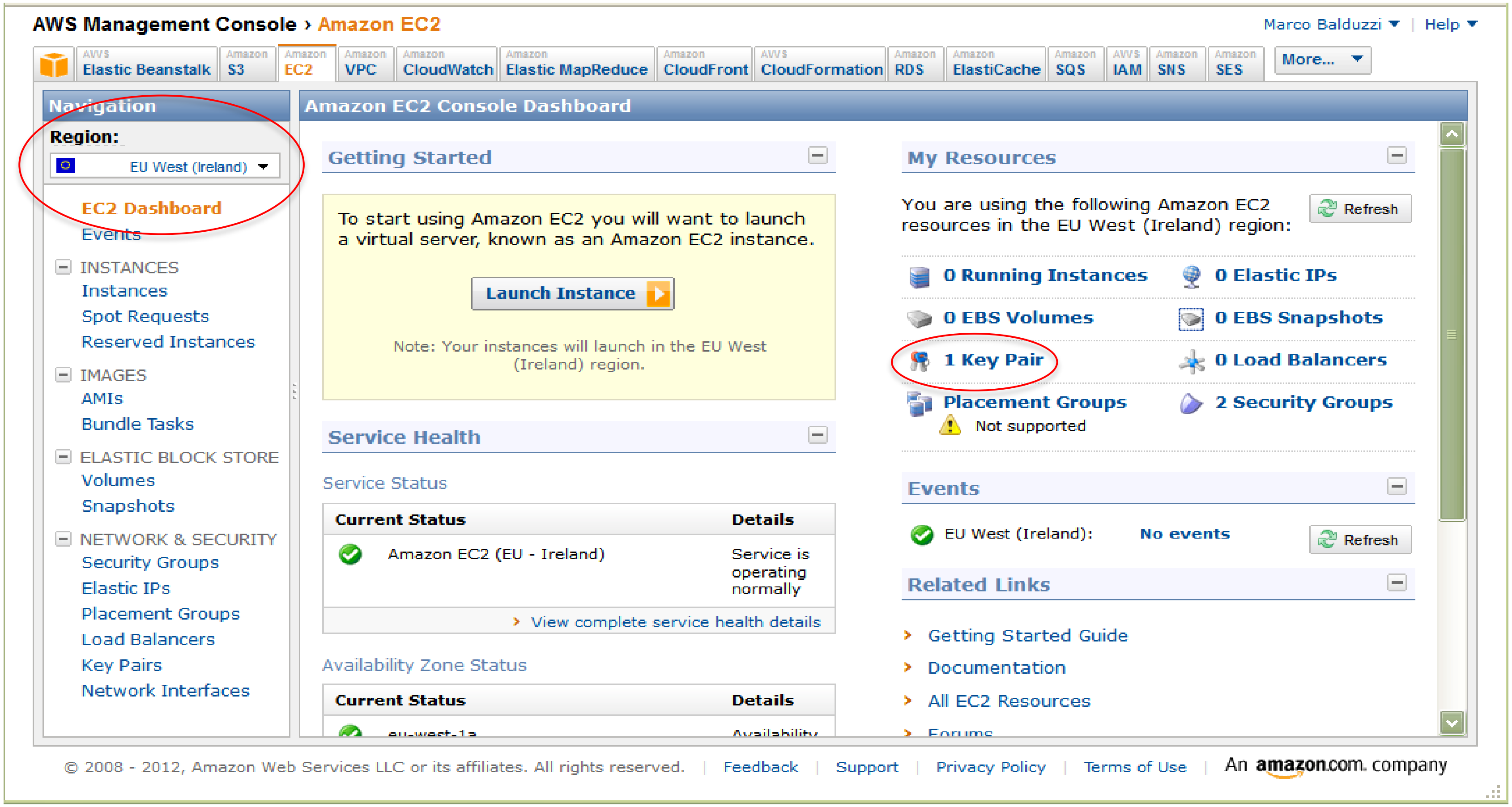
- When an AMI is initiated
  - Hostname is announced
    - e.g. *ec2-IP-region.computer.amazonaws.com*
  - Accessible via SSH (port 22) or Remote Desktop (port 3389)

- Amazon does **not** care about securing the image
  - The maintenance is completely under the **responsibility of the end user**

- User has root privileges, needs to administer system

Usage example [1/3]

- Amazon Web Services (AWS) Management Console



AWS Management Console › Amazon EC2

Marco Balduzzi ▼ | Help ▼

Navigation

Region:

EU West (Ireland) ▼

EC2 Dashboard

Events

⊟ INSTANCES

Instances

Spot Requests

Reserved Instances

⊟ IMAGES

AMIs

Bundle Tasks

⊟ ELASTIC BLOCK STORE

Volumes

Snapshots

⊟ NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Load Balancers

Key Pairs

Network Interfaces

Amazon EC2 Console Dashboard

Getting Started

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance ▶

Note: Your instances will launch in the EU West (Ireland) region.

Service Health

Service Status

Current Status		Details
✓	Amazon EC2 (EU - Ireland)	Service is operating normally

› View complete service health details

Availability Zone Status

Current Status		Details
✓	eu-west-1a	Availability

My Resources

You are using the following Amazon EC2 resources in the EU West (Ireland) region:

Refresh

0 Running Instances	0 Elastic IPs
0 EBS Volumes	0 EBS Snapshots
1 Key Pair	0 Load Balancers
Placement Groups ⚠ Not supported	2 Security Groups

Events

✓ EU West (Ireland): No events

Refresh

Related Links

› Getting Started Guide

› Documentation

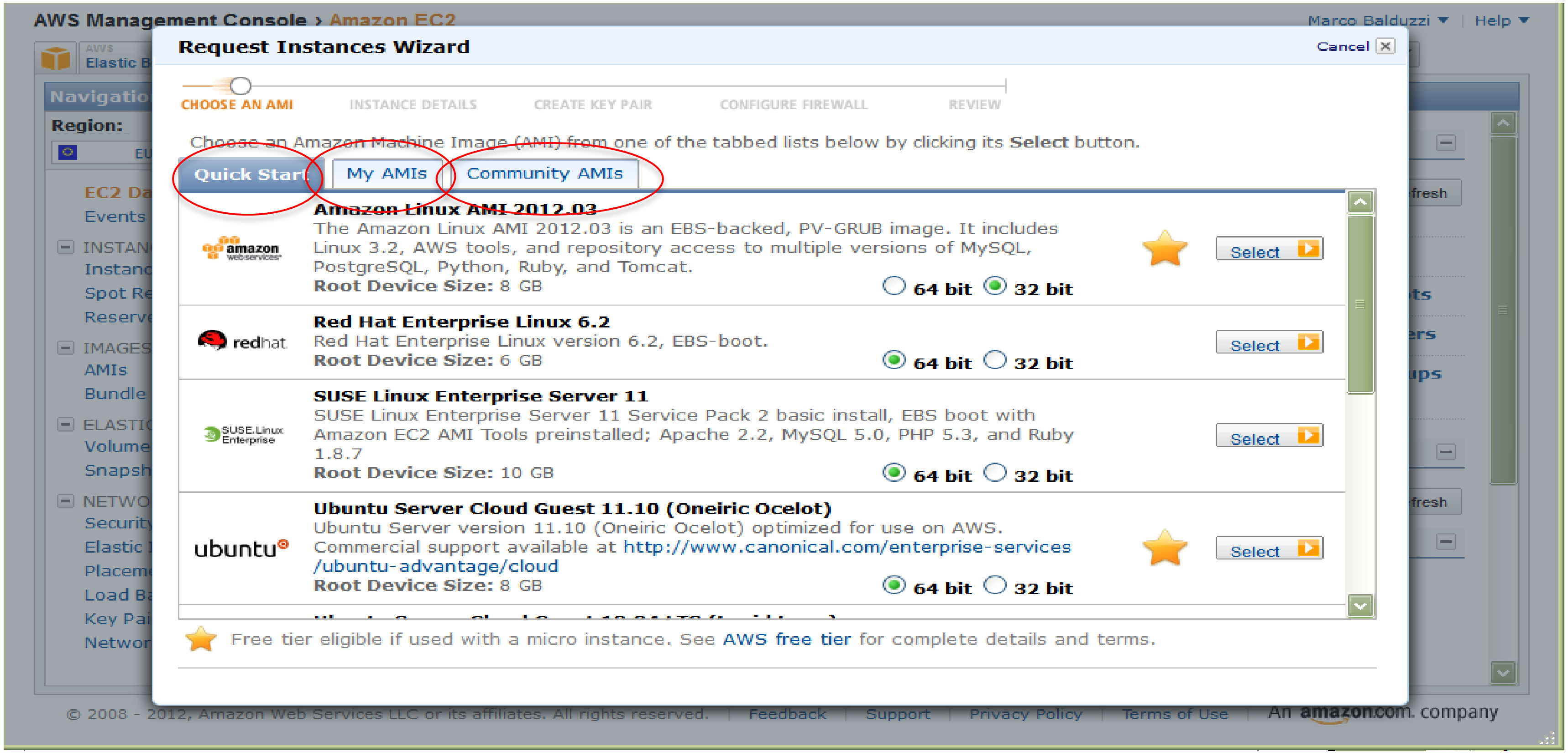
› All EC2 Resources

› Forums

© 2008 - 2012, Amazon Web Services LLC or its affiliates. All rights reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com. company

Usage example [2/3]

- Launch an instance



Request Instances Wizard

Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Amazon Linux AMI 2012.03
The Amazon Linux AMI 2012.03 is an EBS-backed, PV-GRUB image. It includes Linux 3.2, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.
Root Device Size: 8 GB ○ **64 bit** ● **32 bit** ★ Select

Red Hat Enterprise Linux 6.2
Red Hat Enterprise Linux version 6.2, EBS-boot.
Root Device Size: 6 GB ● **64 bit** ○ **32 bit** Select

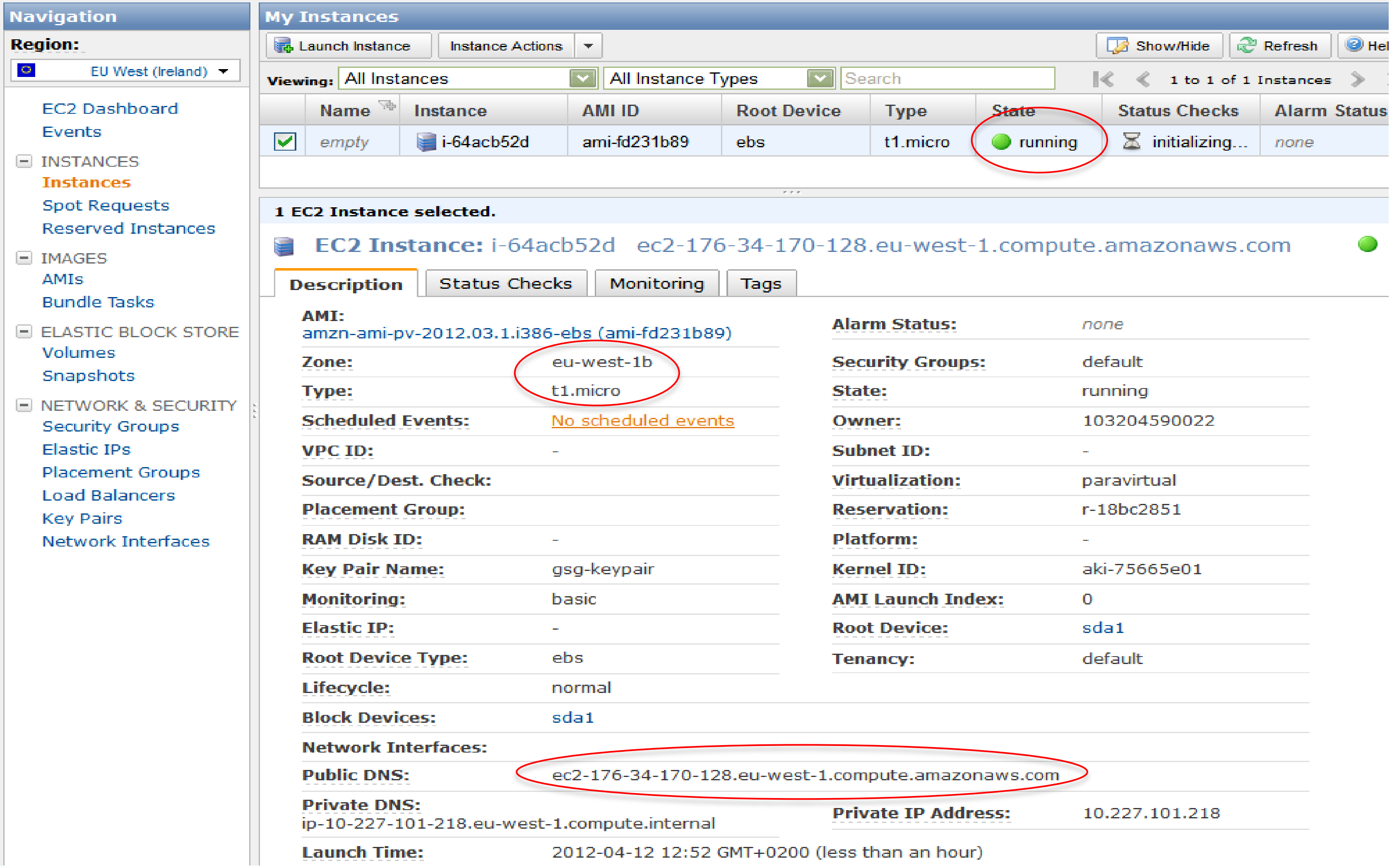
SUSE Linux Enterprise Server 11
SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7
Root Device Size: 10 GB ● **64 bit** ○ **32 bit** Select

Ubuntu Server Cloud Guest 11.10 (Oneiric Ocelot)
Ubuntu Server version 11.10 (Oneiric Ocelot) optimized for use on AWS. Commercial support available at http://www.canonical.com/enterprise-services/ubuntu-advantage/cloud
Root Device Size: 8 GB ● **64 bit** ○ **32 bit** ★ Select

★ Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

© 2008 - 2012, Amazon Web Services LLC or its affiliates. All rights reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com. company

Usage example [3/3]



	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm Status
☑	*empty*	i-64acb52d	ami-fd231b89	ebs	t1.micro	● running	initializing...	*none*

1 EC2 Instance selected.

EC2 Instance: i-64acb52d ec2-176-34-170-128.eu-west-1.compute.amazonaws.com ●

AMI:
amzn-ami-pv-2012.03.1.i386-ebs (ami-fd231b89)

Zone:	eu-west-1b	**Alarm Status:**	*none*
Type:	t1.micro	**Security Groups:**	default
Scheduled Events:	No scheduled events	**State:**	running
VPC ID:	-	**Owner:**	103204590022
Source/Dest. Check:		**Subnet ID:**	-
Placement Group:		**Virtualization:**	paravirtual
RAM Disk ID:	-	**Reservation:**	r-18bc2851
Key Pair Name:	gsg-keypair	**Platform:**	-
Monitoring:	basic	**Kernel ID:**	aki-75665e01
Elastic IP:	-	**AMI Launch Index:**	0
Root Device Type:	ebs	**Root Device:**	sda1
Lifecycle:	normal	**Tenancy:**	default
Block Devices:	sda1		
Network Interfaces:			
Public DNS:	ec2-176-34-170-128.eu-west-1.compute.amazonaws.com		
Private DNS: ip-10-227-101-218.eu-west-1.compute.internal		**Private IP Address:**	10.227.101.218
Launch Time:	2012-04-12 12:52 GMT+0200 (less than an hour)		

Problem definition

- A popular approach is to create, publish and share server images with other users
- Trust model *cloud provider & user* is well-defined
  - i.e., Amazon is not going to hurt you ☺
- What about *image provider & user*?
  - Users can create and share images too… blurry

- Are there any **threats** associated with **renting** images from the **public catalogs** of cloud service providers?
- To which extend?

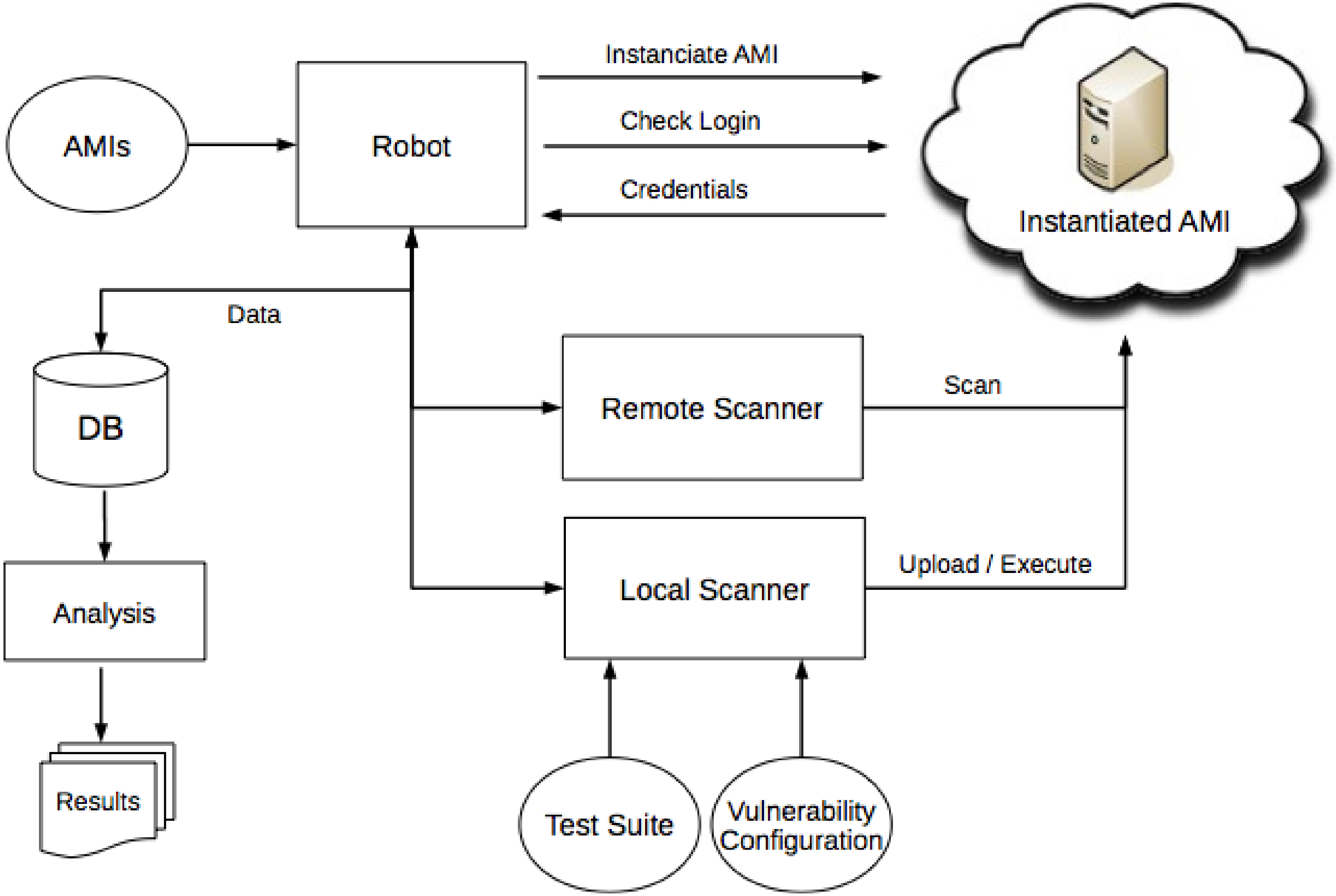
The Threats Landscape

- Securing the Image against **external attacks**
- Securing the Image against **malicious image providers**
- Sanitizing the Image to protect the **privacy** of the image **provider**

Large-scale experiment

- **Automated system** for security analysis and measurement
- **All** public server images provided by Amazon in its four data centers
  - US East, US West, Europe and Asia
- Over a period of 7 months
- Successfully scanned 5,303 AMIs
  - Linux and Windows

SatanCloud


AMIs
Robot
Instanciate AMI
Check Login
Credentials
Instantiated AMI
Data
DB
Remote Scanner
Scan
Local Scanner
Upload / Execute
Analysis
Results
Test Suite
Vulnerability Configuration

Remote Scanner

- It collects information over network

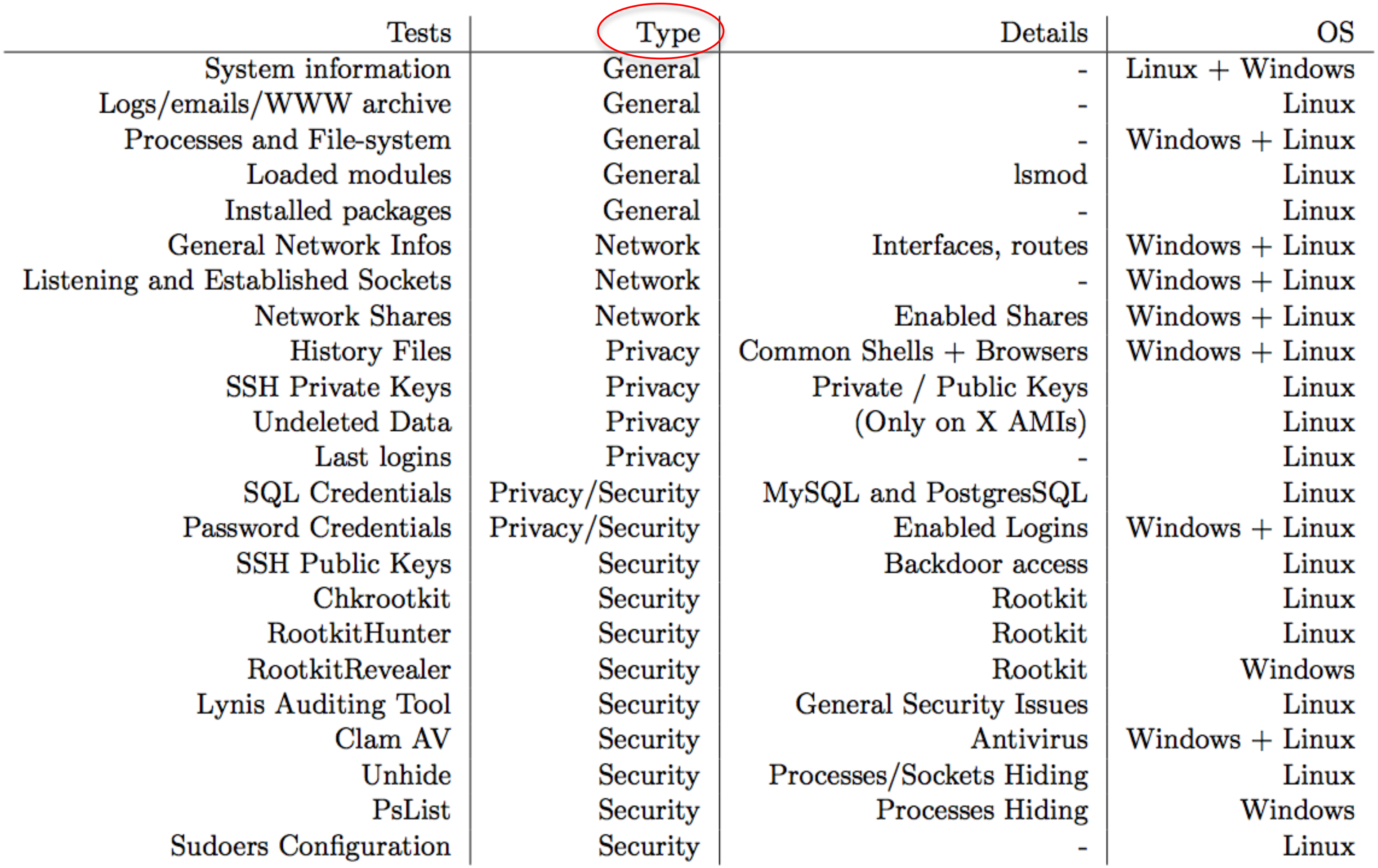
- List the open ports and services (NMap is used)
- The installed web server
- Web modules (name, version)
- Web application (index page)
- Utility? Wait the end of the talk…

Local Scanner, two tasks

- 1. Analyze the AMI for known **vulnerabilities** using the Nessus tool (locally – i.e., precise)

- 2. Upload to AMI and remote execute a **test suite**
- Self-extracting archive that contains 24 tests grouped in 4 categories:
  - General – system information, log files and data collection
  - Network – shared directories, open sockets, running servers
  - Privacy – history files, file-system analysis, forgotten data
  - Security – vulnerable applications, rootkit & malware detection, hidden processes

Overview of Tests We Performed

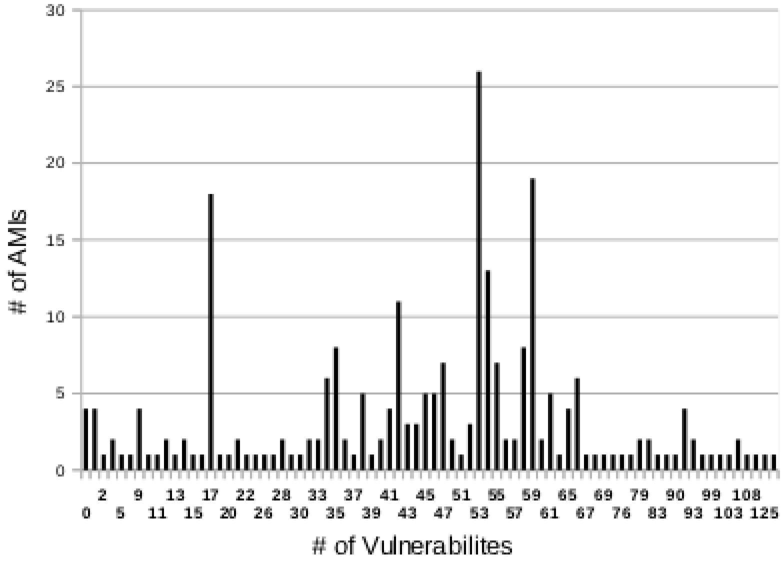


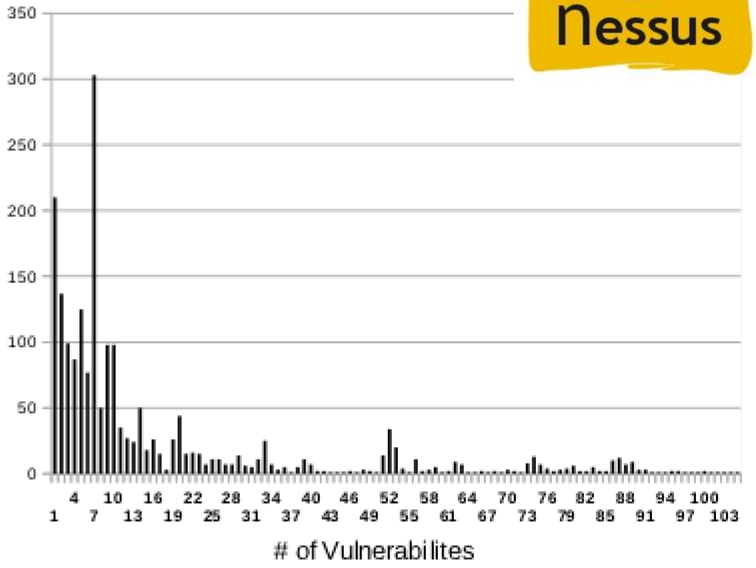
Tests	Type	Details	OS
System information	General	-	Linux + Windows
Logs/emails/WWW archive	General	-	Linux
Processes and File-system	General	-	Windows + Linux
Loaded modules	General	lsmod	Linux
Installed packages	General	-	Linux
General Network Infos	Network	Interfaces, routes	Windows + Linux
Listening and Established Sockets	Network	-	Windows + Linux
Network Shares	Network	Enabled Shares	Windows + Linux
History Files	Privacy	Common Shells + Browsers	Windows + Linux
SSH Private Keys	Privacy	Private / Public Keys	Linux
Undeleted Data	Privacy	(Only on X AMIs)	Linux
Last logins	Privacy	-	Linux
SQL Credentials	Privacy/Security	MySQL and PostgresSQL	Linux
Password Credentials	Privacy/Security	Enabled Logins	Windows + Linux
SSH Public Keys	Security	Backdoor access	Linux
Chkrootkit	Security	Rootkit	Linux
RootkitHunter	Security	Rootkit	Linux
RootkitRevealer	Security	Rootkit	Windows
Lynis Auditing Tool	Security	General Security Issues	Linux
Clam AV	Security	Antivirus	Windows + Linux
Unhide	Security	Processes/Sockets Hiding	Linux
PsList	Security	Processes Hiding	Windows
Sudoers Configuration	Security	-	Linux

Findings

Software vulnerabilities [1/2]

- Nessus performed a precise, **local** scan on the actual software installed
  - Windows, Linux
- We limited the analysis to the **critical** vulnerabilities only





Software vulnerabilities [2/2]

- 98% Windows, 58% Linux AMIs come with critical vulnerabilities

AMIs…	Windows	Linux
with vulnerabilities <= 2 years	145	1,197
with vulnerabilities <= 3 years	38	364
Avg. # vulnerabilities / AMI	46	11

- 87 Debian AMIs come with the now notorious SSH/ OpenSSL vulnerability discovered in May 2008 (i.e., CVE-2008-0166)

Security Risks - Malware

- We used ClamAV to scan systems (850,000 signatures)
- We discovered two infected AMIs, both Windows-based
- Trojan-Spy 50112: key logger, process monitor, and **data leakage** from saved files
- Trojan.Agent 173287: browser **spyware** (IE BHO)
    - Cannot manually confirm the presence
    - The machine got infected during our test experiment?
    - 1h of unpatched execution with no firewall

Security Risks - Unsolicited connections

- Plenty of outgoing connections
- Hard to evaluate each of them
- Two Linux AMIs configured to send the **logs to a remote host**
- syslog-NG

Leftover Credentials

- When user rents AMI, public key needs to be provided
  – Amazon adds this to *authorized_keys* for ssh access
- **Security** Risk: Users could leave key behind and make image public (turn to **backdoor**)
  – Same problem if a user sets password and publishes image

	US East	US West	Europe	Asia	Total
AMIs with leftover credentials	34.75%	8.35%	9.80%	6.32%	21.80%
With password	67	10	22	2	101
With SSH keys	794	53	86	32	965
With both	71	6	9	4	90
Superuser privileges	783	57	105	26	971
User privileges	149	12	12	12	185

- **Privacy** Risk: Passwords can be **cracked** and used by 3rd parties

Privacy risks

- If the image contains sensitive information, these would be available to anybody who is renting the AMI
- Not only customers have a potential risk, but **providers** too
- Accessing credentials, e.g.
  - To login into other servers
  - To start instances “for free”
- Information such as browser history can be used for deanonymization, or social engineering

CONFIDENTIAL

“Forgotten” keys

- We searched the images for forgotten keys
  - `id_dsa` and `id_rsa` for SSH keys
  - `pk-[0-9A-Z]*.pem` and `cert-[0-9A-Z]*.pem` for AWS API keys
- 56 private SSH keys used to login to other machine
  - 54 of which where **not** protected with a passphrase
  - IP of other machines available in the logs :)
- We discovered 67 unprotected AWS API keys
  - Can immediately be used to start images on the cloud at the **expense** of the key’s owner

Shell history

- Shell histories: credentials (usernames and passwords)
  - Automatically inspected *.history, .bash_history, .sh_history*
  - 869 files stored interesting information, 158,354 lines of command history

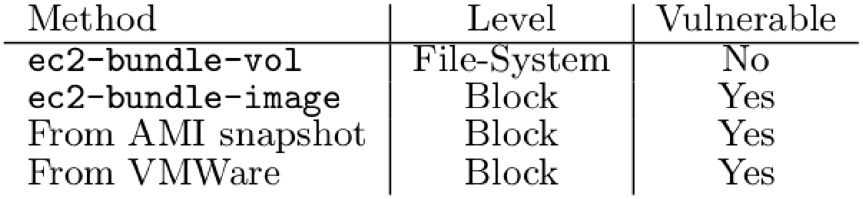
Finding	# Credentials	# Local	# Remote
Amazon RDS	4	0	4
Dynamic DNS	1	0	1
Database Monitoring	7	6	1
MySQL	58	45	13
Web Applications	3	2	1
VNC	1	1	0
Total	74	54	20

$ mysql –u user –p password –h host …

- So if I delete my data then I am fine … ?

Recovery of deleted files [1/3]

- AMIs can be bundled using different methods

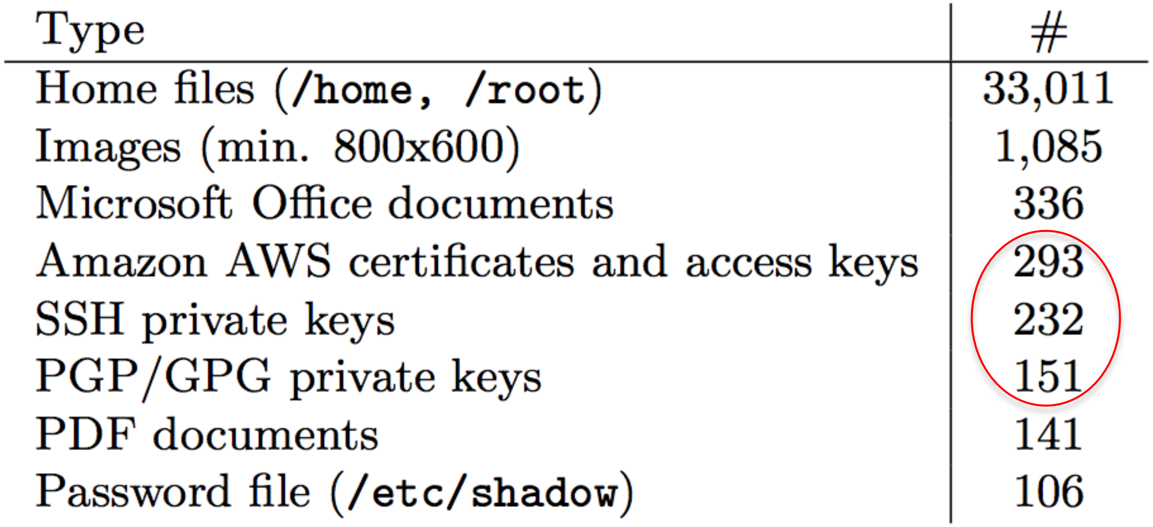


Method	Level	Vulnerable
`ec2-bundle-vol`	File-System	No
`ec2-bundle-image`	Block	Yes
From AMI snapshot	Block	Yes
From VMWare	Block	Yes

- Block-based bundling methods are **vulnerable** to file **undelete attacks**
  - Even if provider deletes files, attacker might still access them
- We randomly selected 1,100 Linux AMIs in 4 regions
- We used `extundelete` to automatically inspect the AMI's filesystem

Recovery of deleted files [2/3]

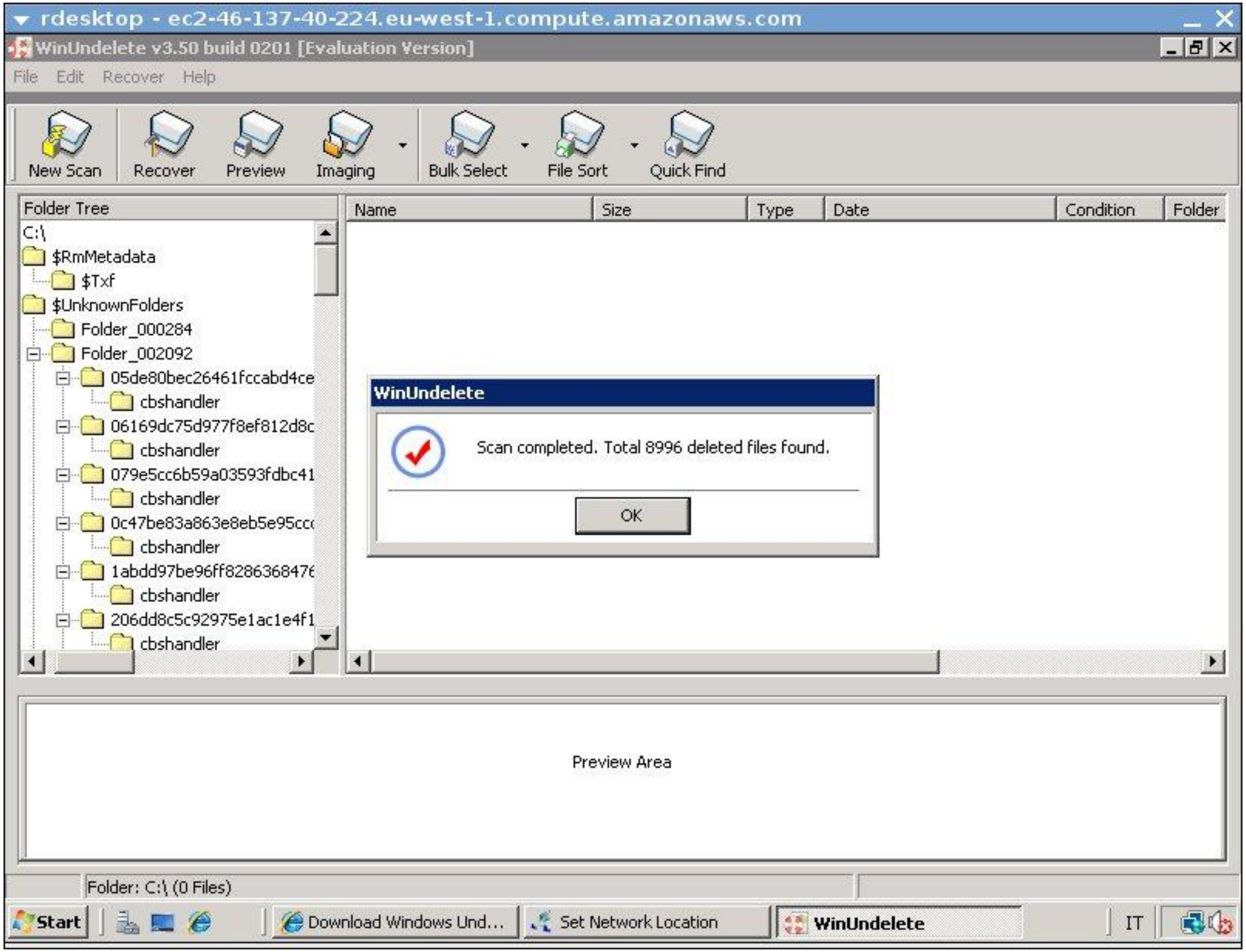
- Were undelete 28GB of data
- We recover files for 98% of the AMIs (6 to 40,000 file per AMI).



Type	#
Home files (`/home, /root`)	33,011
Images (min. 800x600)	1,085
Microsoft Office documents	336
Amazon AWS certificates and access keys	293
SSH private keys	232
PGP/GPG private keys	151
PDF documents	141
Password file (`/etc/shadow`)	106

- Even an official Amazon image (private SSH key!)

Recovery of deleted files [3/3]



Scan completed. Total 8996 deleted files found.

Folder: C:\ (0 Files)

Matching AMIs to Running Instances

- Suppose attacker hides an *ssh* key, how does he **locate** the server?
- Given a running instance on the Amazon EC2 cloud, how to find the corresponding AMI ?
- Perfect solution: **SSH host key**
  - Should be regenerated upon
  - But that is not always the case...
- Approximate solutions
  - Service Banners
  - Web

Experiment

- We scanned the Amazon IP range *(ARIN, RIPE, LAPNIC)*
- 653,401 IPs
- Collected info for 233K running instances

Technique	Instances	Perfect Match	Set of 10 Candidates	Set of 50 Candidates
SSH	130,580	1.65%	6.79%	9.01%
Services	203,563	3.45%	14.91%	31.20%
Web	125,554	4.42%	25.21%	43.74%

Feedbacks and collaboration

- During our experiments we were in **contact** with the AmazonWS Security Team
- 1 - Passwords and public keys
  - Contacted all the clients, released a public bulletin, changed the status of vulnerable AMIs to private
- 2 - Leftover data
  - Released (within 5 days) a tutorial to help customers share public images in a secure manner
- 3 - Recovering deleted data
  - Verified our finding (immediately)
  - AMIs examination (work in progress)

Lessons Learned

- Prepare your **own** image
- Otherwise:
  - Immediately update the software (with the firewall up)
  - Regenerate the SSH host key
  - Delete any user, password, and SSH key
  - Check the configuration files of the services you plan to run
  - Check for suspicious connections
  - … did I tell you to prepare your own image?
- If you plan to release a public image
  - Use a file-based bundle mechanism (or shred any sensitive files)
  - Delete logs and history files

References

- Amazon
  - How to share and use public AMIs in a secure manner
  - Reminder about safely sharing and using public AMIs

- M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, S. Loureiro
  - A Security Analysis of Amazon's Elastic Compute Cloud Service. *In Proceedings of the the 11th edition of the Computer Security track at the 27th ACM Symposium on Applied Computing*

Thanks!

SECLUDIT

iSecLAB

EURECOM Sophia Antipolis

TREND MICRO