Cloud Vulnerability Assessment

Automating Security in the Cloud

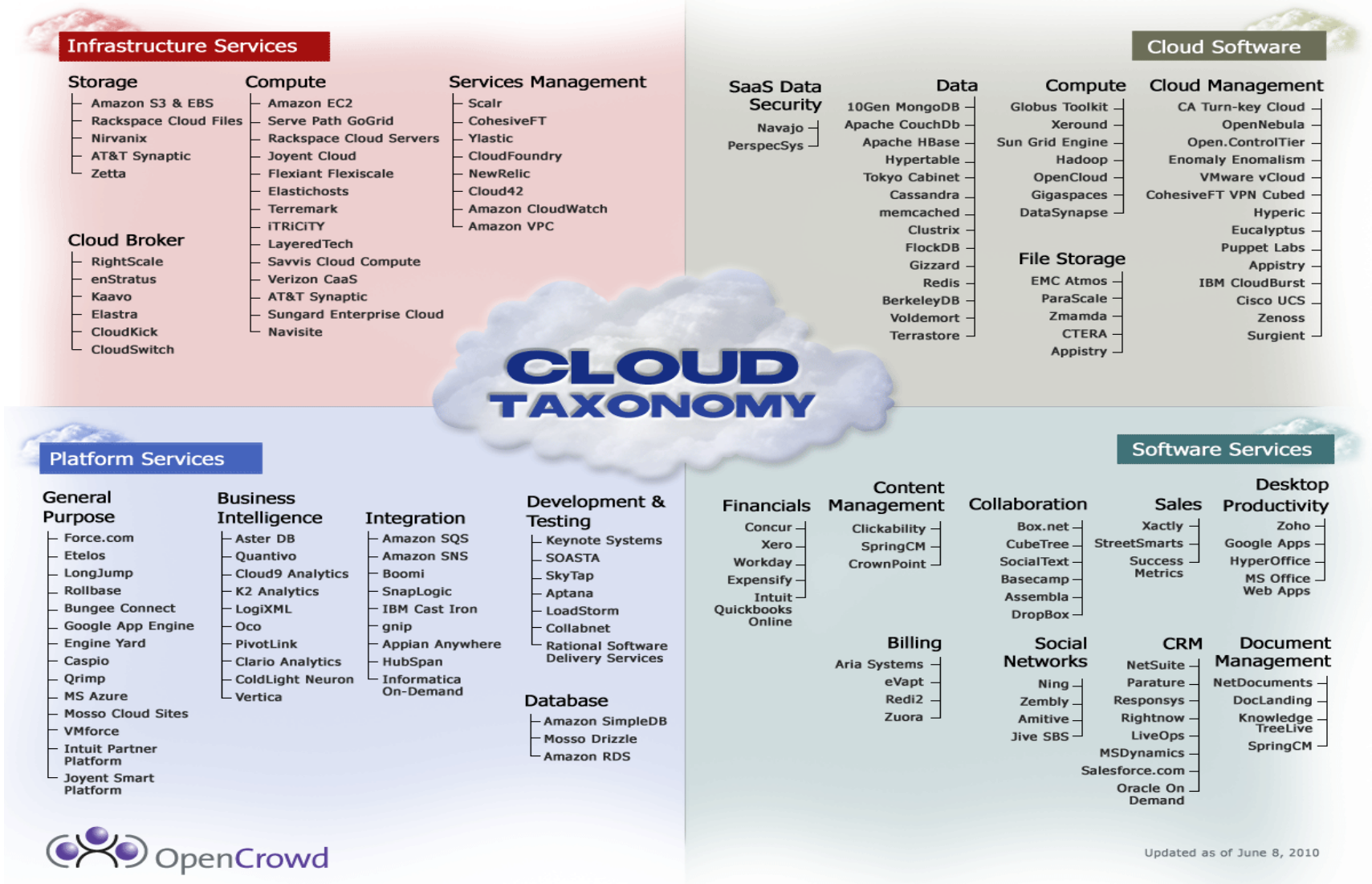
Presented by:

Mika Ayenson, Andre Guerlain

Our Contributions

- Researched cloud security solutions and potential tools for examining different types of vulnerabilities.
- Designed and built a open-source framework for performing vulnerability assessments within the cloud.
- Created modules to scan for vulnerabilities and later exploit discovered vulnerabilities.
- Built a database in which to store the results in an easily accessible format.
- Tested the modules for functionality and performance.

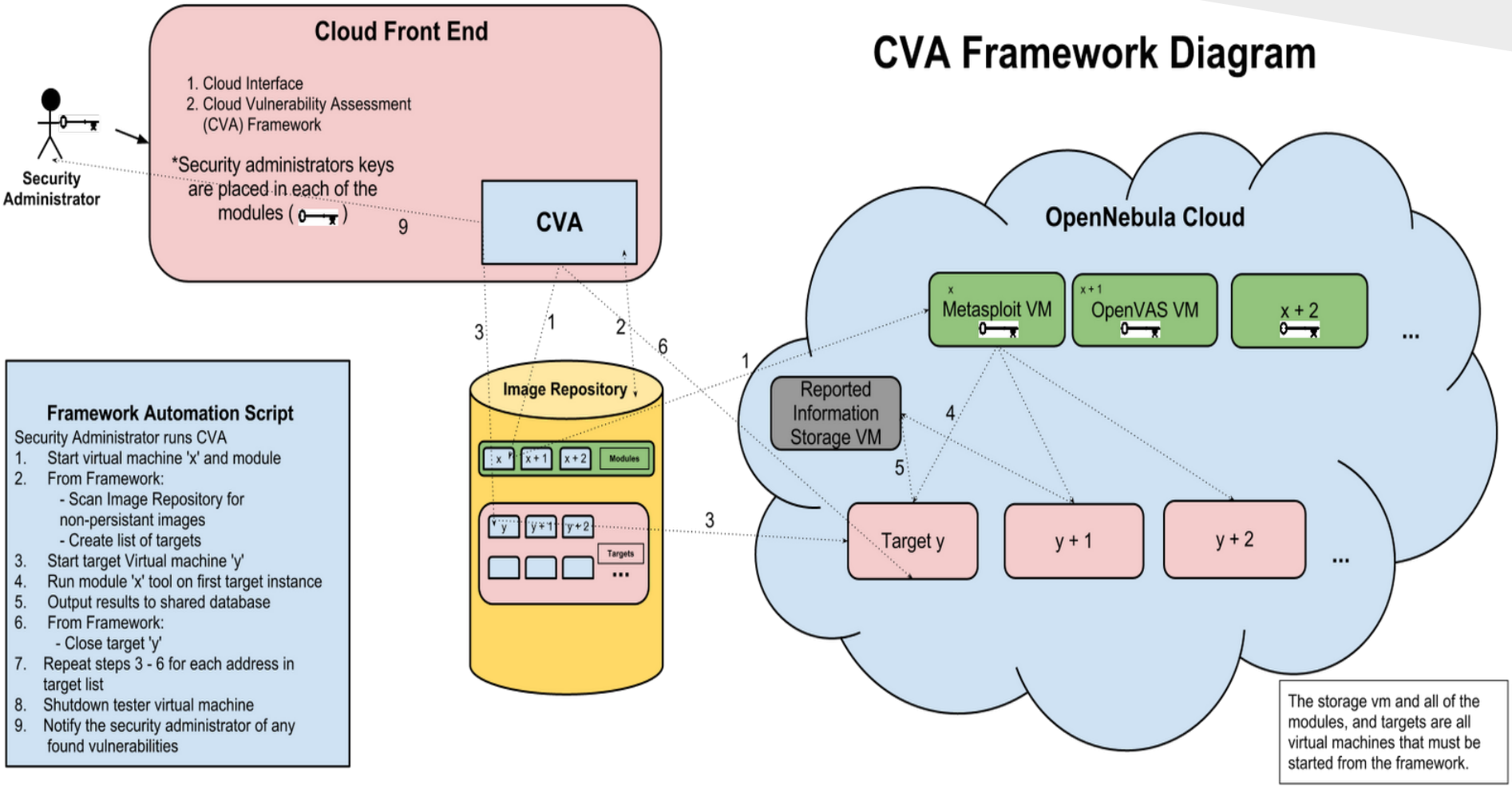
CSA Guide


Infrastructure Services
Storage
Amazon S3 & EBS
Rackspace Cloud Files
Nirvanix
AT&T Synaptic
Zetta
Compute
Amazon EC2
Serve Path GoGrid
Rackspace Cloud Servers
Joyent Cloud
Flexiant Flexiscale
Elastichosts
Terremark
iTRiCiTY
LayeredTech
Savvis Cloud Compute
Verizon CaaS
AT&T Synaptic
Sungard Enterprise Cloud
Navisite
Services Management
Scalr
CohesiveFT
Ylastic
CloudFoundry
NewRelic
Cloud42
Amazon CloudWatch
Amazon VPC
Cloud Broker
RightScale
enStratus
Kaavo
Elastra
CloudKick
CloudSwitch
Cloud Software
SaaS Data Security
Navajo
PerspecSys
Data
10Gen MongoDB
Apache CouchDb
Apache HBase
Hypertable
Tokyo Cabinet
Cassandra
memcached
Clustrix
FlockDB
Gizzard
Redis
BerkeleyDB
Voldemort
Terrastore
Compute
Globus Toolkit
Xeround
Sun Grid Engine
Hadoop
OpenCloud
Gigaspaces
DataSynapse
File Storage
EMC Atmos
ParaScale
Zmamda
CTERA
Appistry
Cloud Management
CA Turn-key Cloud
OpenNebula
Open.ControlTier
Enomaly Enomalism
VMware vCloud
CohesiveFT VPN Cubed
Hyperic
Eucalyptus
Puppet Labs
Appistry
IBM CloudBurst
Cisco UCS
Zenoss
Surgient
CLOUD TAXONOMY
Platform Services
General Purpose
Force.com
Etelos
LongJump
Rollbase
Bungee Connect
Google App Engine
Engine Yard
Caspio
Qrimp
MS Azure
Mosso Cloud Sites
VMforce
Intuit Partner Platform
Joyent Smart Platform
Business Intelligence
Aster DB
Quantivo
Cloud9 Analytics
K2 Analytics
LogiXML
Oco
PivotLink
Clario Analytics
ColdLight Neuron
Vertica
Integration
Amazon SQS
Amazon SNS
Boomi
SnapLogic
IBM Cast Iron
gnip
Appian Anywhere
HubSpan
Informatica On-Demand
Development & Testing
Keynote Systems
SOASTA
SkyTap
Aptana
LoadStorm
Collabnet
Rational Software Delivery Services
Database
Amazon SimpleDB
Mosso Drizzle
Amazon RDS
Software Services
Financials
Concur
Xero
Workday
Expensify
Intuit Quickbooks Online
Content Management
Clickability
SpringCM
CrownPoint
Billing
Aria Systems
eVapt
Redi2
Zuora
Collaboration
Box.net
CubeTree
SocialText
Basecamp
Assembla
DropBox
Social Networks
Ning
Zembly
Amitive
Jive SBS
Sales
Xactly
StreetSmarts
Success Metrics
CRM
NetSuite
Parature
Responsys
Rightnow
LiveOps
MSDynamics
Salesforce.com
Oracle On Demand
Desktop Productivity
Zoho
Google Apps
HyperOffice
MS Office Web Apps
Document Management
NetDocuments
DocLanding
Knowledge TreeLive
SpringCM
OpenCrowd
Updated as of June 8, 2010

CSA Guide (Cont.)

- **Interoperability and portability**
- Virtualization
- Security for cloud computing
- **Security as a Service**
- Information management and data security
- Application security
- Application penetration testing for the cloud

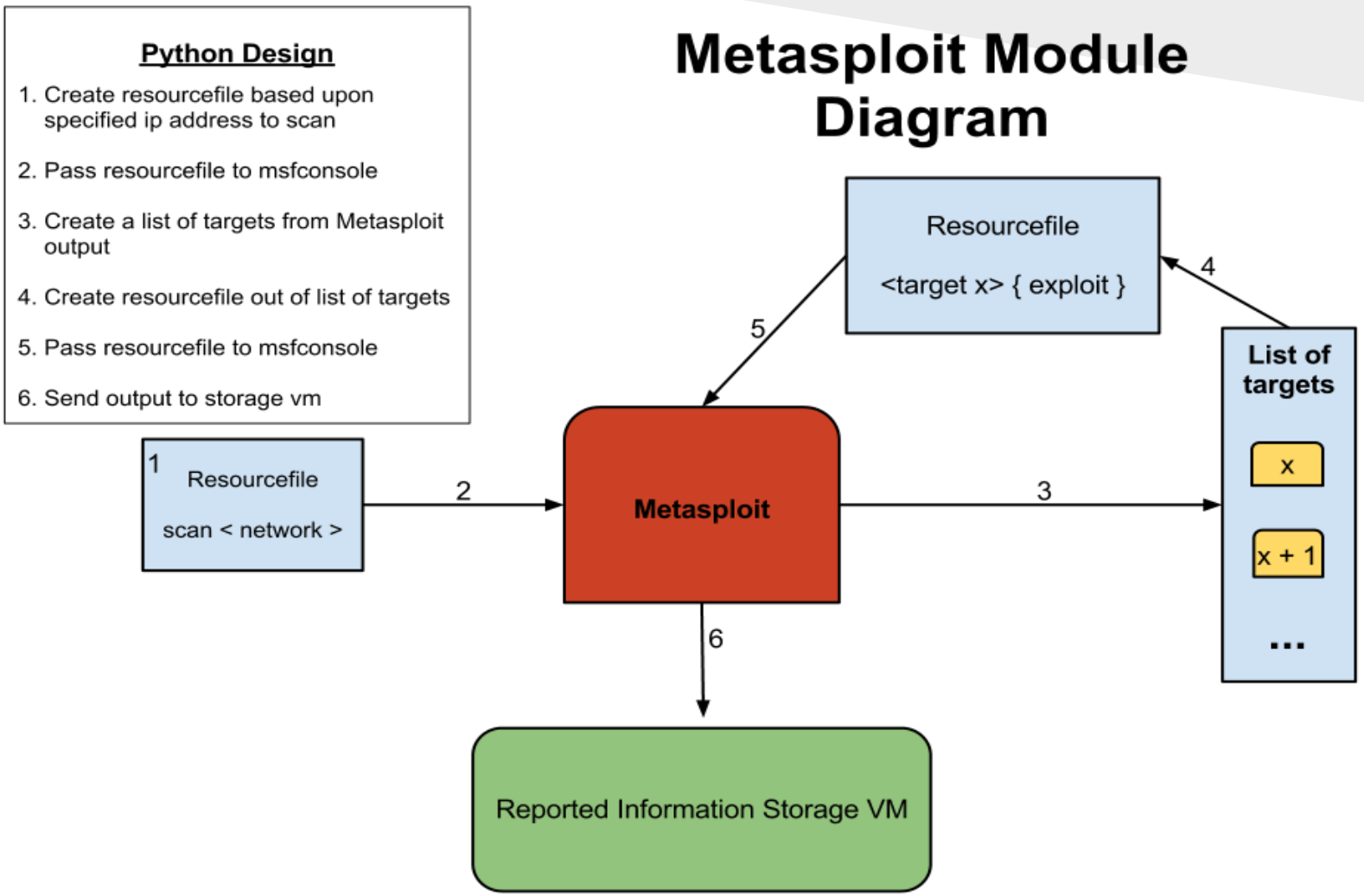
Cloud Vulnerability Assessment



Sploit

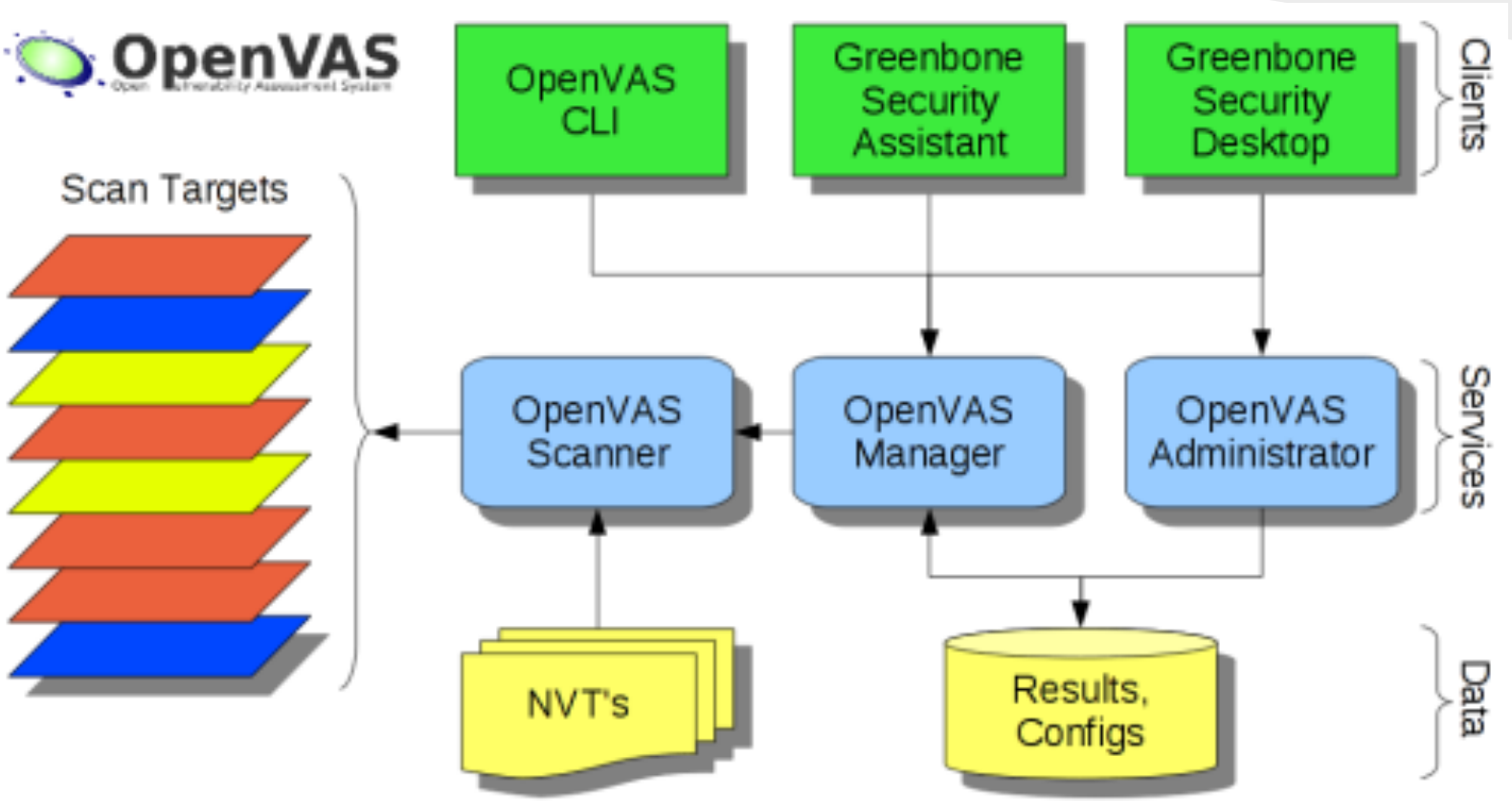
Python Design

1. Create resourcefile based upon specified ip address to scan
2. Pass resourcefile to msfconsole
3. Create a list of targets from Metasploit output
4. Create resourcefile out of list of targets
5. Pass resourcefile to msfconsole
6. Send output to storage vm



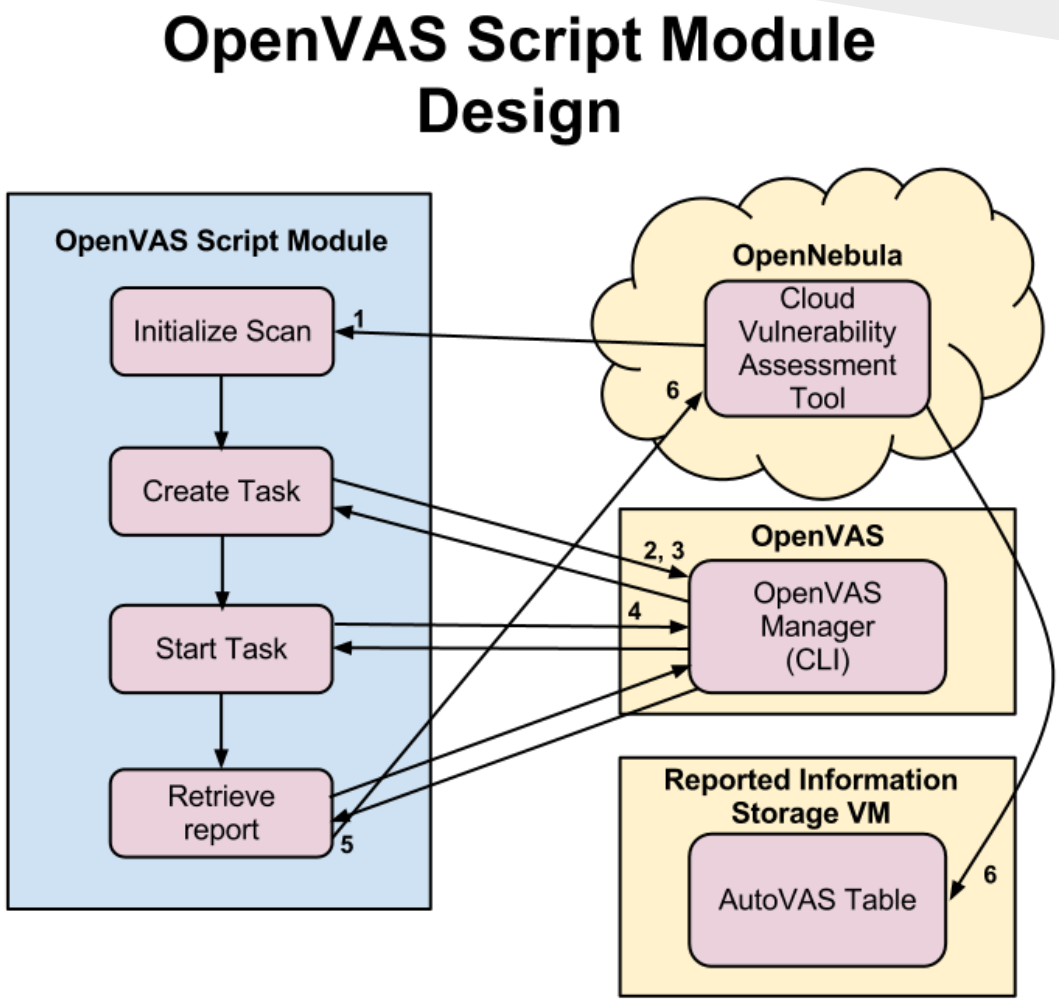
[2]

OpenVAS

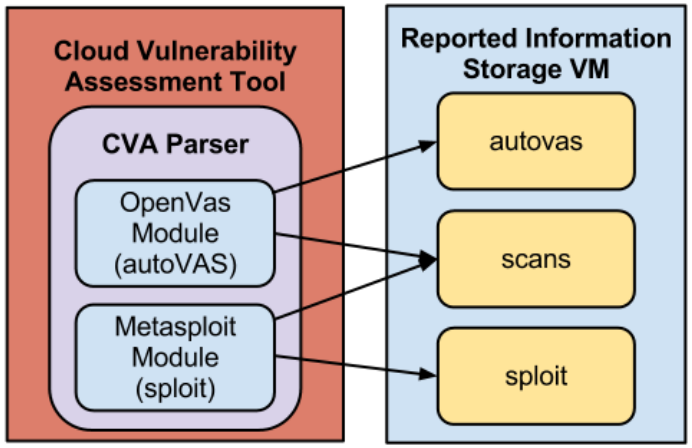

OpenVAS
OpenVAS CLI
Greenbone Security Assistant
Greenbone Security Desktop
Clients
Scan Targets
OpenVAS Scanner
OpenVAS Manager
OpenVAS Administrator
Services
NVT's
Results, Configs
Data

[3]

autoVAS


OpenVAS Script Module Design
OpenVAS Script Module
Initialize Scan
Create Task
Start Task
Retrieve report
1
2, 3
4
5
6
OpenNebula
Cloud Vulnerability Assessment Tool
OpenVAS
OpenVAS Manager (CLI)
Reported Information Storage VM
AutoVAS Table

Information Storage System



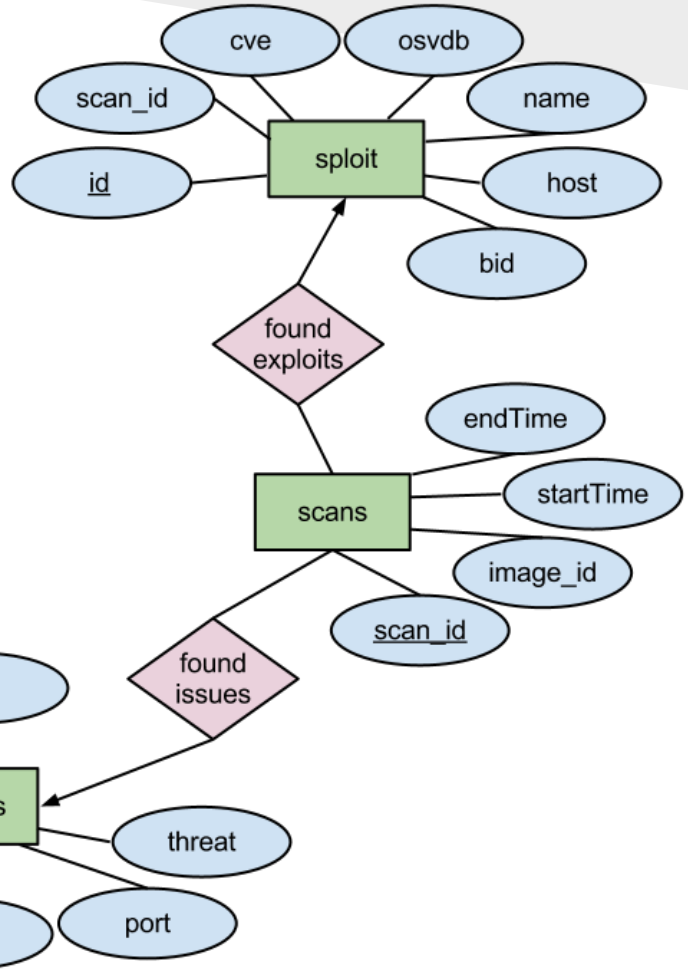
The OpenVAS module stores the vulnerabilities that it finds in the corresponding autovas table.

The Metasploit module stores the exploitable vulnerabilities that it finds in the corresponding sploit table.

The CVA script within the Cloud Front End parses the result files, before storing the data within the database VM.

Both modules create a tuple in the scan table each time they are run on a target machine.

The scans have a one to many relationship with the results in the sploit and autovas tables



Database Tables

```
mysql> show tables;
+-----------------+
| Tables_in_CVAdb |
+-----------------+
| autovas         |
| scans           |
| sploit          |
+-----------------+
3 rows in set (0.00 sec)

mysql> describe scans;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| scan_id   | int(11)     | NO   | PRI | NULL    | auto_increment |
| image_id  | varchar(20) | YES  |     | NULL    |                |
| startTime | varchar(30) | YES  |     | NULL    |                |
| endTime   | varchar(30) | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

```
mysql> describe sploit;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int(11)      | NO   | PRI | NULL    | auto_increment |
| host    | varchar(20)  | YES  |     | NULL    |                |
| cve     | varchar(40)  | YES  |     | NULL    |                |
| osvdb   | varchar(40)  | YES  |     | NULL    |                |
| bid     | varchar(40)  | YES  |     | NULL    |                |
| name    | varchar(100) | YES  |     | NULL    |                |
| scan_id | int(11)      | YES  |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)

mysql> describe autovas;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int(11)      | NO   | PRI | NULL    | auto_increment |
| scan_id | int(11)      | YES  |     | NULL    |                |
| nvt     | varchar(150) | YES  |     | NULL    |                |
| oid     | varchar(60)  | YES  |     | NULL    |                |
| threat  | varchar(40)  | YES  |     | NULL    |                |
| port    | varchar(40)  | YES  |     | NULL    |                |
| cve     | varchar(40)  | YES  |     | NULL    |                |
| bid     | varchar(40)  | YES  |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

Machines Tested

ID : 169 NAME : debian 6.0 TYPE : OS	ID : 168 NAME : CVA_DB TYPE : OS	ID : 166 NAME : edgi-testwms TYPE : OS
ID : 165 NAME : edgi-testvoms TYPE : OS	ID : 164 NAME : edgi-testui TYPE : OS	ID : 163 NAME : edgi-testboinc TYPE : OS
ID : 146 NAME : Metasploitable TYPE : OS	ID : 143 NAME : BackTrack 5r2 TYPE : OS	ID : 141 NAME : Volatile-Datablock-16GB TYPE : DATABLOCK
ID : 136 NAME : WinXP-Base TYPE : OS	ID : 130 NAME : SPEQULOS_BOINC TYPE : OS	ID : 99 NAME : bridge_w_metajob TYPE : OS
ID : 111 NAME : SPEQULOS TYPE : OS	ID : 66 NAME : M3S TYPE : OS	ID : 65 NAME : SALMon TYPE : OS

Results

Functional and Performance Tests:

32 public images

AutoVAS Module:

1. Total time 4:17:02, an average of 8:02 per scan.
2. Found 560 results
    a. High: 24
    b. Medium: 36
    c. Low: 184
    d. Log: 316

Sploit Module:

1. Total time 2:08:26, an average of 4:49 per scan
2. Opened 1 session via PHP injection

Results (Cont.)

AutoVAS then Sploit:

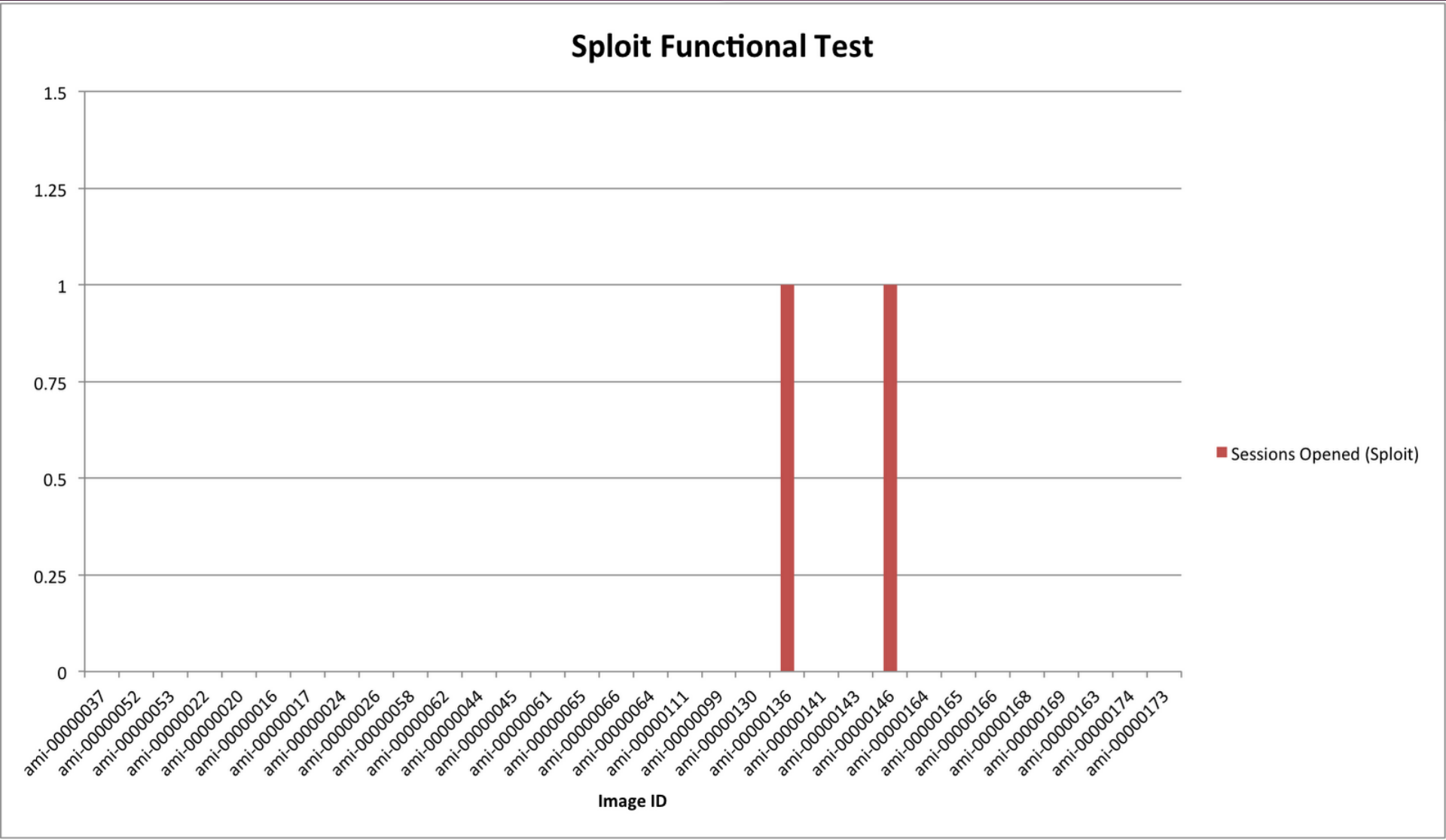
1. Total Time 6:43:04, an average of 6:18 per scan

2. AutoVAS
    a. Found 648 results
3. Sploit
    a. Opened 2 sessions on targets using,
        i. PHP injection, and
        ii. Buffer overflow allowing arbitrary code execution

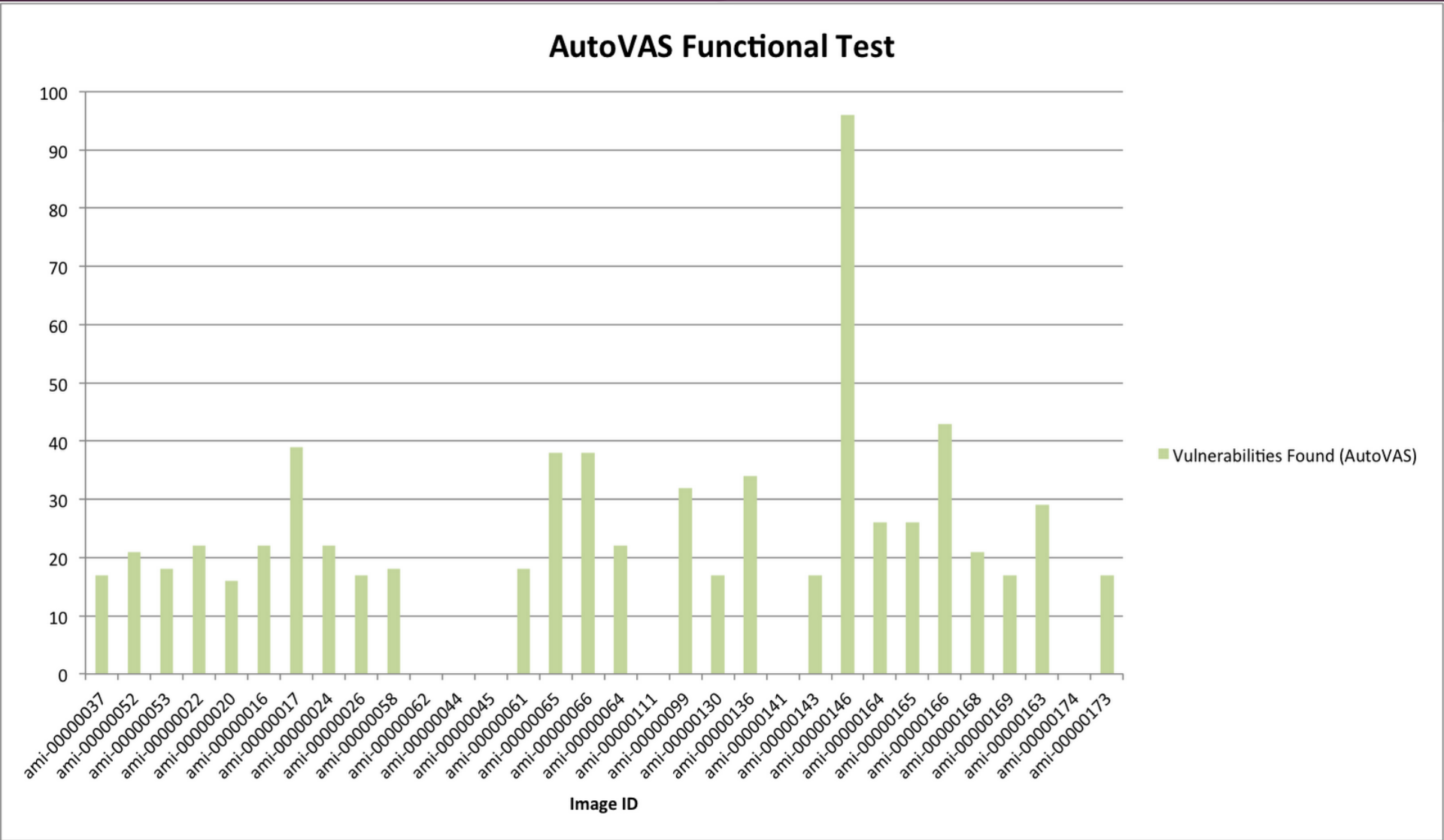
Sploit then AutoVAS:

1. Total Time 7:21:13, an average of 6:54 per scan
2. AutoVAS
    a. Found 719 results

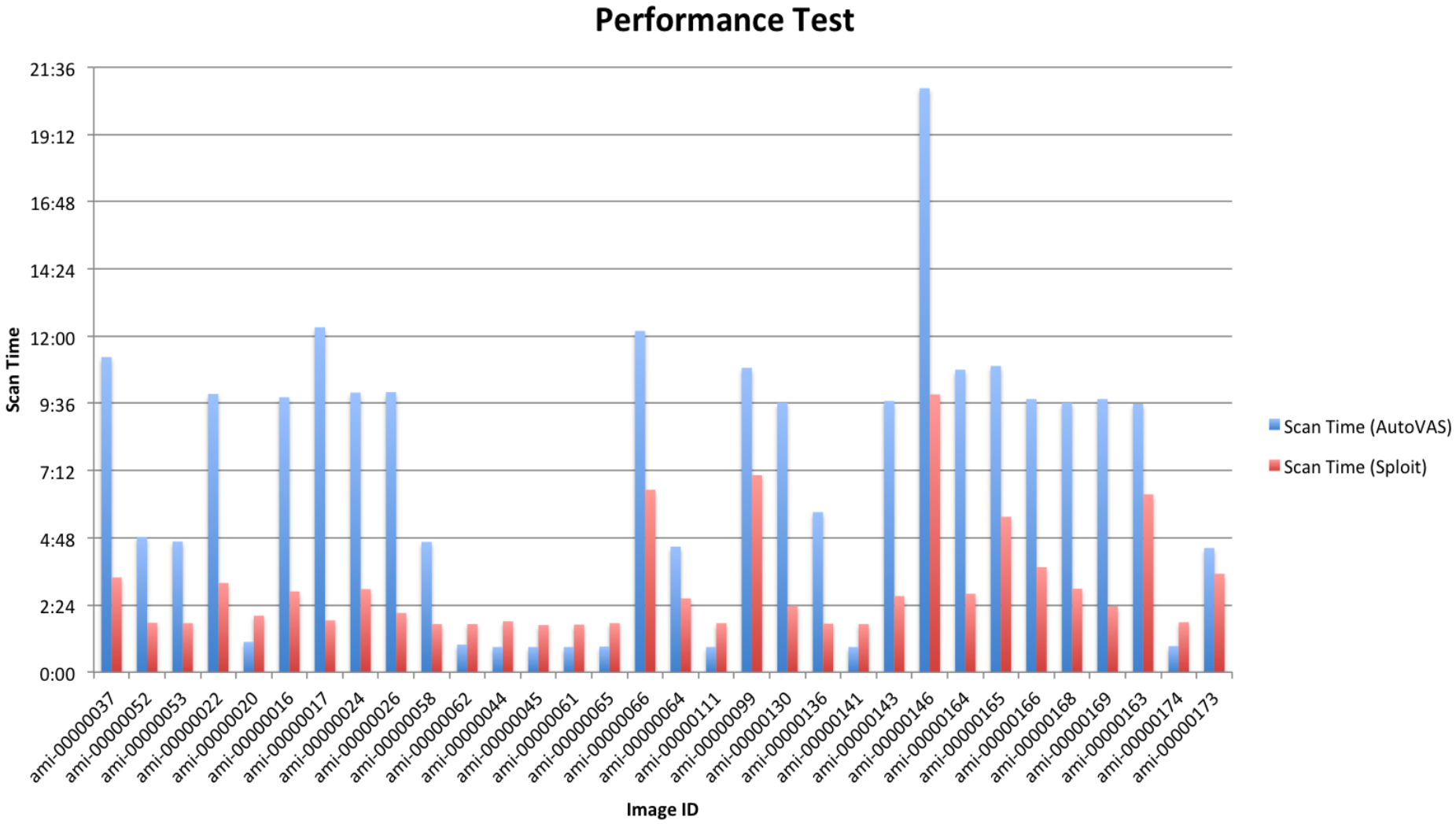
Functional Test sploit



Functional Test AutoVAS



Performance Test



Conclusion

Our Cloud Vulnerability Assessment System Works!

Future Work

- Automatic Exploit and Vulnerability Patches
- More Modules
- Encryption and Key Management
- cvaFrame privledges

Check out:

https://github.com/RDaemon5/cvaFrame_framework

Acknowledgements

Gábor Sárközy, MQP Advisor

Sándor Ács, Project Advisor and SZTAKI contact

Márk Gergely, Co-Project Advisor and SZTAKI contact

Peter Kacsuk, SZTAKI Department Head

Stanley Selkow, MQP Co-Advisor

Róbert Lovas, SZTAKI Program Director

Péter Kotcauer, SZTAKI colleague

MTA-SZTAKI

Worcester Polytechnic Institute

Acknowledgements? (cont.)

Kösönöm szepen!

Questions?

References

[1] Cloud Security Alliance ed.. Security Guidance For Critical Areas Of Focus In Cloud Computing V3.0. CSA, 2011. Web 2 April 2012. <https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf>

[2] OpenVAS (). OpenVAS Framework. Retrieved 31 March 1990 from OpenVAS:http://www.openvas.org/about.html

[3] Rapid 7 (). Metasploit Framework. Retrieved 31 March 2012 from Rapid 7:http://www.metasploit.com/

[4] OpenNebula (). About the OpenNebula.org Project. Retrieved 31 March 2012 from

OpenNebula:http://opennebula.org/about:about