SECRETASASERVICE

Key Management for the Open Cloud

Jarret Raim & Matt Tesauro

ABOUT US

ACADEMIC

TRINITY UNIVERSITY
SAN ANTONIO • 1869

LEHIGH UNIVERSITY

DEVELOPER

SwRI®

SECURITY CONSULTANT

DENIM GROUP℠

APPLICATION SECURITY

SECURITY PRODUCTS

OWASP BOARD MEMBER

OWASP LIVE CD

OWASP WTE

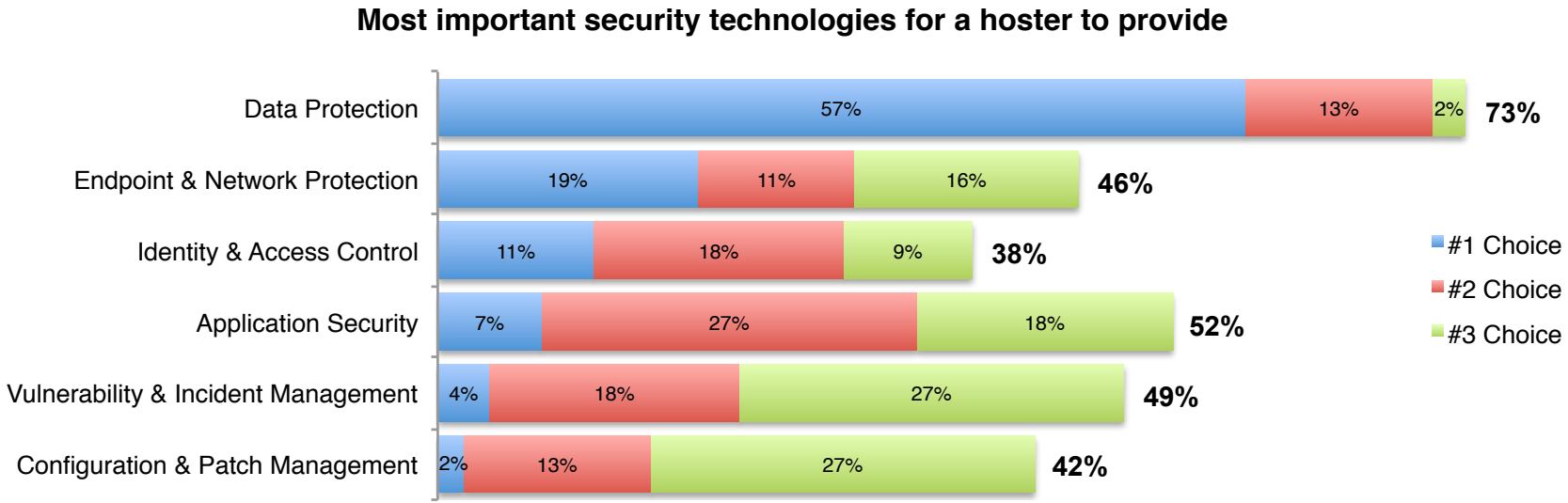
RACKER SINCE ‘11

PRODUCT SECURITY

HACKING THE RACK

What do customers mean by security?

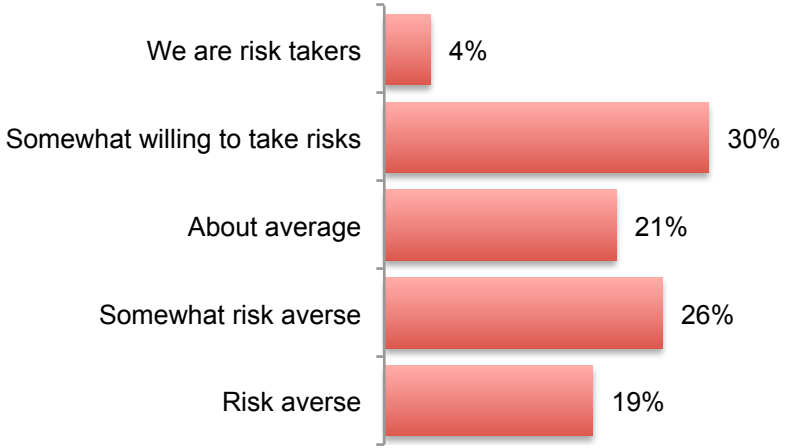
SECURITY TAXONOMY



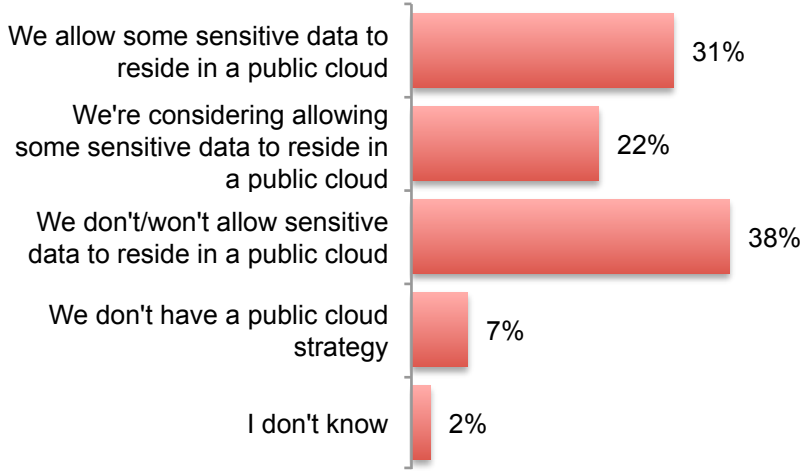
Customers don't want to give us their data

RISKY BUSINESS

Organization risk tolerance

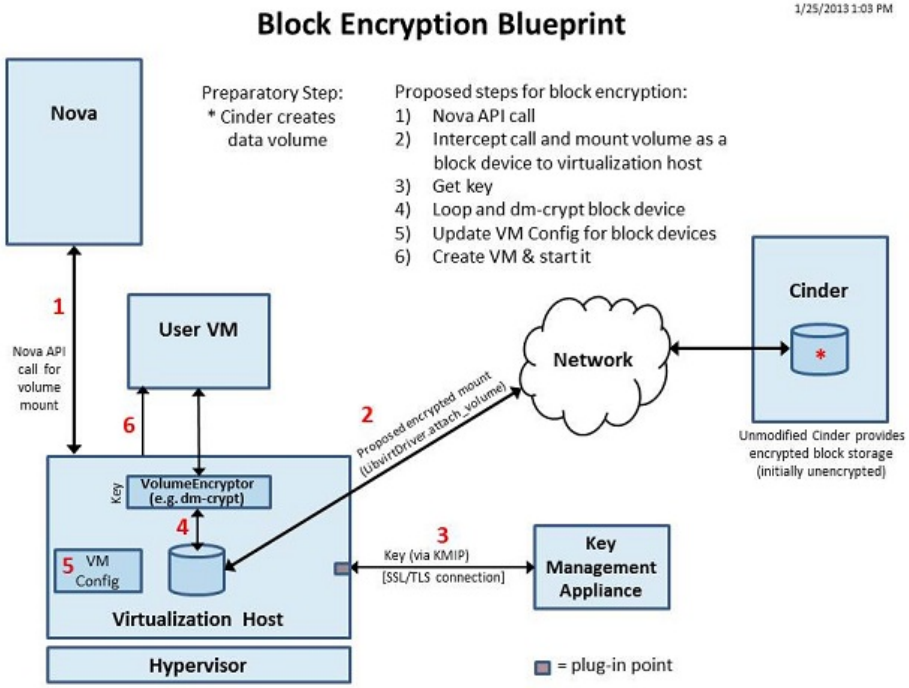


Cloud strategy regarding sensitive data



Cinder and Swift already pushing forward

CURRENTPLANS



PROTOCOL SUPPORT

Must support different protocols so that multiple products can integrate to the same system.

IDENTITY

Must support standard Keystone authentication methods.

MULTI-TENANT

Must support all tenants for a Cloud in the same system with guaranteed isolation.

AUDITING & COMPLIANCE

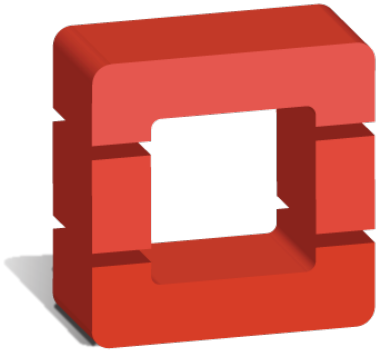
Must support auditing & logging to support various compliance regimes.

FREE & OPEN SOURCE

Must support for all environments, public and private.

Every OpenStack project has encryption needs

FUTUREPLANS



openstack™

CLOUD SOFTWARE

CINDER, SWIFT & GLANCE

Encrypted files at rest.

RED DWARF

Encrypted databases and tables.

QUANTUM

SSL Certificates and VPN keys.

NOVA

SSH keys, encrypted file systems.

KEYSTONE

Encrypted metadata, user level keys

DON'T FORGET THE CUSTOMERS

Customer applications running on Cloud have a different, but overlapping, set of needs from OpenStack services.

MULTI-CLOUD INTEROPERABILITY

Customers want to be able to store their keying material in a different physical & legal environment than their data. We must support multi-cloud use cases and key sharing.

EASY INTEGRATION

Many legacy applications were not designed with advanced key management in mind. Customers need easy ways to retrofit existing applications, integrate new ones and connect vendor solutions.

CENTRALLY MANAGED

Key management is easy to get wrong. Customers need an easy to manage solution with optional expert assistance in configuration and monitoring.

IMPROVED SECURITY & COMPLIANCE

Most customers have compliance requirements to meet. We must support those needs while enabling real security improvements.


1
C

Look familiar?

BADCODE

```
public class CryptHelper {
  private static final String ALOGRITHM = "PBEWithMD5AndTripleDES";

  // Salt
  static byte[] salt = { (byte) 0xc1, (byte) 0xa3, (byte) 0x28, (byte)
0x1c, (byte) 0x7b, (byte) 0xc9, (byte) 0x1e, (byte) 0x9e };

  static PBEKeySpec pbeKeySpec = new PBEKeySpec("chamber of
                                                secrets".toCharArray());

  public byte[] encrypt(String cleartext) { … }
  public String decrypt(byte[] ciphertext) { … }
}
```

GitHub Search - Greatest hacker resource ever

BADDEFAULTS

branch: **master** | **Files** | Commits | Branches 1 | Tags

[redacted] / config / initializers / **secret_token.rb**

[redacted] 3 months ago Rails & Bootstrap

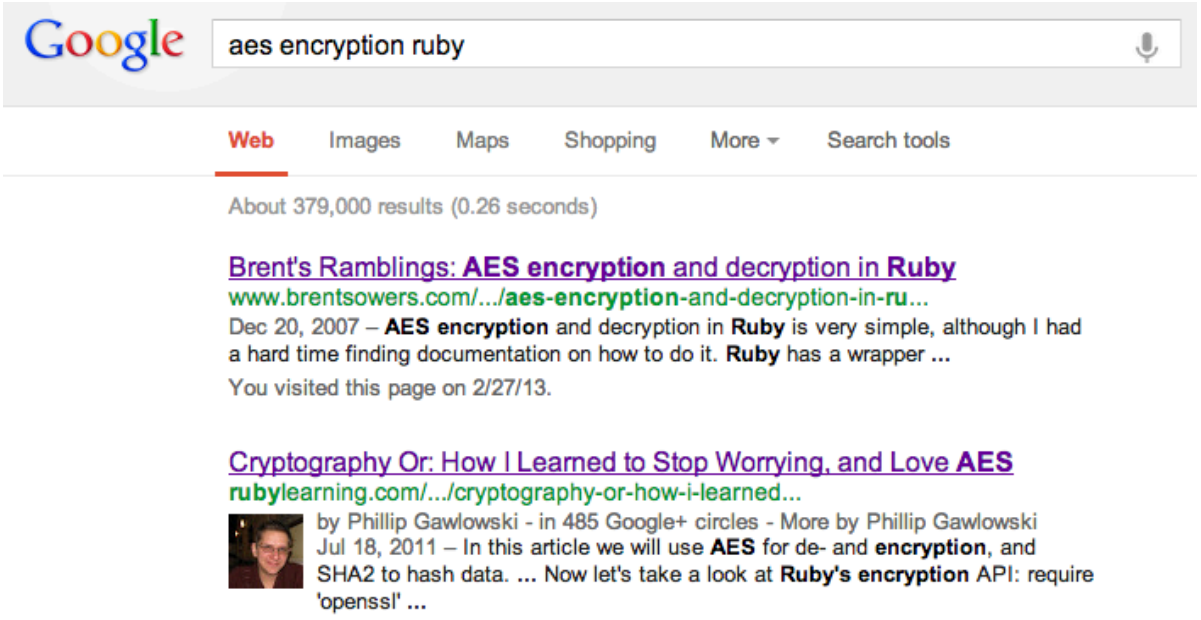
0 contributors

file | 8 lines (6 sloc) | 0.496 kb | Edit | Raw | Blame | History

```
1  # Be sure to restart your server when you modify this file.
2
3  # Your secret key for verifying the integrity of signed cookies.
4  # If you change this key, all old signed cookies will become invalid!
5  # Make sure the secret is at least 30 characters and all random,
6  # no regular words or you'll be exposed to dictionary attacks.
7  [redacted]::Application.config.secret_token = '[redacted]
```

First 10 results on Google - all bad

BADADVICE



- HARDCODED KEYS
- HARDCODED ALGORITHMS
- BAD CYPHER BLOCK TYPES
- OLD
- NULL & HARDCODED IV'S

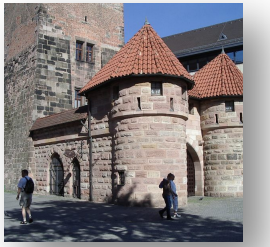
Open source key management

INTRODUCING CLOUDKEEP

https://github.com/cloudkeep

That's right, it's castle themed

MEETTHEKEEPS



BARBICAN

ReST API

Barbican is the main ReST API providing secret storage, provisioning, lifecycle management auditing and reporting. It is written in Python using Falcon, Oslo and following OpenStack standards.

POSTERN

Agent

Postern is the agent that provides access to secret material. It is currently planned to be implemented in Go, but we may use the Rackspace Cloud Monitoring agent framework (Virgo).

PALISADE

Web UI

Palisade is a client side JavaScript MVC application that provides a web interface for Barbican. It is written in AngularJS and can be delivered from the Barbican API server or as a Chrome plugin.

KEEP

Command line client

Keep is a python based command line client similar to python-novaclient. It is most useful for server maintenance, troubleshooting and development.

DESIGN PRINCIPALS

1. Provide a central key-store capable of distributing keying material to all types of deployments including ephemeral Cloud instances.
2. Support reasonable compliance regimes through reporting and auditability.
3. Application adoption costs should be minimal or non-existent.
4. Build a community and ecosystem by being open-source and extensible.
5. Improve security through sane defaults and centralized management of key policies.
6. Out of band communication mechanism to notify and protect sensitive assets.
7. Use OpenStack tools, processes, libraries and design patterns to ensure easy integration into the ecosystem.

ATTACK OF THE PLUGINS

HARDWARE SECURITY MODULES (HSM)

API

VeriSign Secured™

INTERNAL & EXTERNAL CERTIFICATE AUTHORITIES

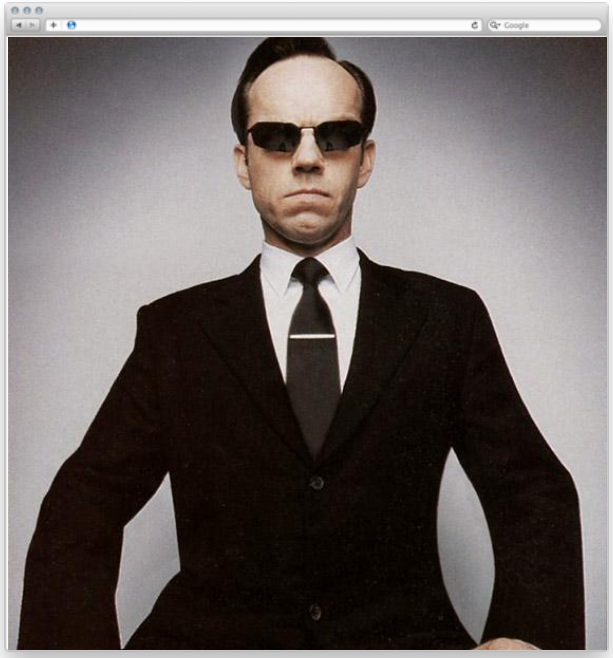
CLOUD LOAD BALANCERS

AUTOMATIC PROVISIONING TARGETS

DATA STORAGE BACKENDS

Oh god, not another agent...

THE AGENT



LEGACY APPLICATION INTEGRATION

The agent presents a FUSE file system to allow applications easy integration options.

ENFORCES POLICIES

Each secret has a set of policies that dictate its use. These policies are mostly enforced by the agent.

KEYSTONE INTEGRATED

The agent uses keystone for identity, pairing and policy management.

OUT OF BAND COMMUNICATION

The agent communicates with the API to represent real-time data about secret usage.

Reusable block of configuration governing secret access

EXAMPLE POLICY

```
{
  "uuid": "01fb57ff-058c-4d68-...",
  "name": "Polyglot policy",
  "max_key_accesses": 1,
  "minutes_available_after_reboot": 10,

  "events": {
    "log_sinks": ["api", "syslog"],
    "file_path": "/var/log/postern.log",
    "allow_override": false,
    "allow_panic": true,
  },

  "executable": {
    "minutes_available_after_restart": 10,
    "name": "mysqld",
    "path": "/usr/bin",
    "owner": "mysql",
    "group": "mysql",
    "hash": "44aea8f32fa3f1f4..."
  },

  "filesystem": {
    "directory_name": "chamber",
    "owner": "root",
    "group": "root",
    "listable": false
  },
}
```

The keying material

EXAMPLESECRET

```
{
  "uuid": "e2b633c7-fda5-…",
  "cacheable": false,
  "expiration": "2014-02-28T19:14:44.180394",
  "secret": "b7990b786ee9659b43ec5…",
  "secret_type": "application/aes-256-cbc",

  "filesystem": [
    {
      "name": "configuration_key",
      "presentation": "file",
      "permissions": "300",
      "owner": "root",
      "group": "root"
    }
  ]
}
```

DEMOTIME

~QUESTIONS?~