Security/OSSA-Metrics

< Security

Work in progress

Inhaltsverzeichnis

Introduction

The OpenStack Security Group suggests that when OpenStack Security Advisories are created by the VMT use the following metrics to score the potential impact of vulnerabilities on OpenStack Deployments

As with all scoring systems this will not be universally applicable but will provide basic guidance to the severity of each vulnerability.

The OSSG has adapted the DREAD (https://www.owasp.org/index.php /Threat_Risk_Modeling) metric as a basis for OpenStack vulnerability impact assessment. We adapted each of the scoring categories to better reflect the impact of a vulnerability in a cloud context.

Public vs Private Cloud

One of the difficulties we face when trying to determine the impact of a vulnerability in OpenStack is understanding how it affects different deployment types. An argument can be made that authenticated users on a private cloud could be more trusted than in a public cloud. However, in designing this threat metric we assert that you cannot confidently trust all employees using a private cloud any more than you can trust users of a public cloud - an assertion somewhat validated by the regular identification of malicious insiders (http://en.wikipedia.org/wiki/Insider_threat) as one of the biggest threats to any organisation[1].

Scoring does not replace experience

Scoring with DREAD and classifying with STRIDE are both subjective. The calibration section below should help the VMT and other stakeholders to make reasoned judgments about the impact of each vulnerability and maintain consistency between the ratings of multiple issues. STRIDE and DREAD should be used to help frame the conversations around what the vulnerability can be used to do and the impact of it being exploited.

STRIDE

We suggest that the VMT classify vulnerabilities in line with STRIDE (https://www.owasp.org/index.php/Threat_Risk_Modeling). This does not have to be a complex task. It just provides a simple and established framework for describing how a vulnerability might be leveraged by an attacker.

- Spoofing: An attacker can through some security failure assume the identity of another user
- Tampering: An attacker can change user or system data without appropriate authorization
- Repudiation: An attacker can take actions that are not recorded / logged, even though they should be
- Information Disclosure: An attacker is able to extract information from the system that should not be available to them
- Denial of Service: Attacker is able to deny service to valid users
- Escalation: Attacker is able to elevate privileges or access beyond that which was assigned to them

In our scoring below we use STRIDE as content for the 'Potential' header.

DREAD

DREAD scores five categories, which are summed together and divided by five, the result is a score from 0-10 where 0 indicates no impact and 10 is the worst possible outcome:

Risk = (DAMAGE + REPRODUCIBILITY + EXPLOITABILITY + AFFECTED USERS + DISCOVERABILITY) / 5

Damage Potential

- If the vulnerability is exploited, how much damage will be caused?
  - 0 = Nothing
  - 3 = Individual user data is compromised, affected or availability denied
  - 5 = All individual tenant data is compromised, affected or availability denied
  - 7 = All tenant data is compromised, affected or availability denied
  - 7 = Availability of a specific cloud controller components/service is denied
  - 8 = Availability of all cloud controller components is denied
  - 9 = Underlying cloud management and infrastructure data is compromised or affected
  - 10 = Complete system or data destruction, failure or compromise

Reproducibility

- How reliably can the vulnerability be exploited?
  - 0 = Very hard or impossible, even for administrators. The vulnerability is unstable and statistically unlikey to be reliably exploited
  - 5 = One or two steps required, tooling / scripting readily available
  - 10 = Unauthenticated users can trivially and reliably exploit using only a web browser

Exploitability

- How difficult is the vulnerability to exploit?
  - 0 = N/A We assert that every vulnerability is exploitable, given time and effort. All scores should be 1-10
  - 1 = Even with direct knowledge of the vulnerability we do not see a viable path for exploitation
  - 2 = Advanced techniques required, custom tooling. Only exploitable by authenticated users
  - 5 = Exploit is available/understood, usable with only moderate skill by authenticated users

- 7 = Exploit is available/understood, usable by non-authenticated users
- 10 = Trivial - just a web browser

Note: In this context, authentication refers to OpenStack users. Users on compute nodes, interacting with virtualised applications are considered to be non-authenticated. A hypervisor breakout would be considered a non-authenticated attack.

Affected Users

- How many users will be affected?
  - 0 = None
  - 5 = Specific to a given project
  - 10 = All users impacted

Discoverability

- How easy is it to discover the threat, to learn of the vulnerability (By convention this is set to 10 even for privately reported vulnerabilities)
  - 0 = Very hard to impossible to detect even given access to source code and privilege access to running systems
  - 5 = Can figure it out by guessing or by monitoring network traces
  - 9 = Details of faults like this are already in the public domain and can be easily discovered using a search engine
  - 10 = The information is visible in the web browser address bar or in a form

Describing Vulnerability Scores

We expect the impact of a vulnerability to be described in the following way:

Potential for: Tampering, Escalation		
Category	**Score**	**Rationale**
Damage	6	Significant Disruption
Reproducability	8	Code path is easily understood, condition exists as standard
Exploitability	2	Very hard to exploit without specific conditions
Affected Users	8	All cloud compute users
Discoverability	10	Discoverability always assumed to be 10
DREAD SCORE: 31/5 = 6.2 - Important, fix as a priority		

[Potentially some nice graph]

Score Categories / Recommendations

The seemingly natural next step here would be to catagorize vulnerabilities based on the DREAD score, perhaps 0-3 would be "Trivial, fix in next release" whereas 8-10 may be "Critical, fix immediately". It's not 100% clear that catagorizing in this way should be the responsibility of the VMT. By producing a DREAD score the VMT has told deployers what the vulnerability is likely to affect and how severely it will do so as well as providing a mechanism to compare one vulnerability against another.

These scores will also provide value for security analysis over time, better catagorization of vulnerabilities (through STRIDE) and scoring of impact (through DREAD) will allow the community to view which particular areas of design and implementation seem to be the worst from a security standpoint.

Calibration

Here we take a number of recent OpenStack Security Advisories and attempt to apply the above metrics. Scoring will always be subjective but the hope is we can use these previous vulnerabilities to first tune the scoring for each category and use them later to validate scores for new vulnerabilities.

OSSA 2014-038

- Title: List instances by IP results in DoS of nova-network
- Link: https://bugs.launchpad.net/ossa/+bug/1358583
- Importance Assigned: Medium

Summary

On a customer install which has approximately 500 VMs in the system, running the following will hang: nova list --ip 199

What will happen afterwards is that the nova-network process will stop responding for a while, a trace shows that it's receiving a huge amount of data.

Dread Score

Potential for: Denial of Service		
Category	**Score**	**Rationale**
Damage	5	Significant Disruption to API/Management
Reproducability	8	Code path is easily understood, condition exists as standard
Exploitability	8	Exploitable with cURL, Authentication Required
Affected Users	4	All Nova API users
Discoverability	10	Discoverability always assumed to be 10
DREAD SCORE: 35/5 = 7 Critical		

Discussion

Did the dread score match the perceived threat from the vulnerability?

OSSA 2013-012

- Title: Nova fails to verify image virtual size
- Link: https://bugs.launchpad.net/nova/+bug/1177830
- Importance Assigned: Critical

Summary

Nova did not implement checking for the virtual size of a qcow2 image used as ephemeral storage for instances. It is therefore possible for a user to create an image which has a large virtual size and then proceed to fill the virtual disk, and consume all available disk on the host node

Dread Score

Potential for: Denial of Service		
Category	**Score**	**Rationale**
Damage	7	Denial of service on the local node and all tenants on it
Reproducability	5	Code path is easily understood, condition exists when using QCOW2
Exploitability	6	Authorized user able to provision compute resources
Affected Users	6	All Nova compute users on the node
Discoverability	10	Discoverability always assumed to be 10
DREAD SCORE: 34/5 = 6.8 - Important, fix as a priority		

Discussion

Did the dread score match the perceived threat from the vulnerability?

OSSA 2014-029

- Title: Catalog replacement allows reading config
- Link: https://bugs.launchpad.net/ossa/+bug/1354208
- Importance Assigned: Medium

Summary

Anyone that can create endpoints in Keystone can then read any value out of the Keystone config file. Some of the values in the config file are passwords, or the admin token for example.

Dread Score

Potential for: Information Disclosure		
Category	**Score**	**Rationale**
Damage	10	If you get the admin token you can potentially delete all users, create users with extra permissions, etc.
Reproducability	7	If you have authority to create endpoints it's easy to do.
Exploitability	5	Users must be granted the authority to create endpoints, which is unlikely.
Affected Users	5	Specific to Keystone.
Discoverability	10	Discoverability always assumed to be 10
DREAD SCORE: 37/5 = 7.4 - Important, fix as a priority		

Discussion

Did the dread score match the perceived threat from the vulnerability?

1. [1] (http://www.cert.org/insider-threat/research/case-analysis-and-best-practices.cfm)