Main page
Recent changes
Server admin log (Prod)
Server admin log (RelEng)
Deployments
SRE/Operations Help
Incident status

Cloud VPS & Toolforge
Cloud VPS documentation
Toolforge documentation
Request Cloud VPS project
Server admin log (Cloud VPS)

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Toolforge webservices are in the final stages of migrating to the toolforge.org domain.
Please help us clean up older documentation referring to tools.wmflabs.org!

Incident documentation/20140203-LVS

< Incident documentation

copy/pasted from Mark's email to Ops list

Contents [hide]

Summary

TL;DR : A bad Puppet change regarding sysctl & RP filtering broke LVS balancers and went undetected until we hit today’s large scale outage.

Timeline

On Monday February 3 2013 around 10:15 UTC, while investigating other network issues, I decided to do fail overs on our LVS load balancers in eqiad. I wanted to update the clocks on these systems to aid my investigation, as they are explicitly not running ntpd for performance reasons. System reboots seemed an easy solution, as I could use the opportunity to upgrade their kernels with the latest security patches as well. We have extensive failover in place for or LVS load balancing infrastructure, so system reboots are normally not disruptive and not a problem. I started with a passive (standby) balancer, lvs1004, which showed no issues during the reboot.

Then around 10:55 UTC I moved on to the active LVS balancer, lvs1001 and lvs1002, which are responsible for balancing all public traffic. I failed over traffic to lvs1004 and lvs1005, and verified that the LVS services were still working by telnetting to some of the TCP ports. However, soon after I commenced the reboot, I received Icinga and Nimsoft pages, for HTTPS services in particular. I attributed this to an issue we’ve had with the way we’ve configured HTTPS services on separate HTTPS terminator systems in esams, and expected this problem to be present in eqiad as well. (However, this is a configuration we’ve already moved away from recently and wasn’t the cause of this issue.) The rebooting LVS server returned a few minutes later, and diagnostic routing output and outside TCP port connects indicated that the LVS services were functional.

Well after lvs1001 and lvs1002 had returned from reboot and restored BGP sessions, around 11:15 UTC, it became clear that LVS still wasn’t working correctly. HTTP services seemed to function, but HTTPS users were still receiving 503 errors. I suspected that the internal LVS service IPs, used by the SSL terminators to communicate to the HTTP Varnish caches, were not working while the public service IPs did work. Some internal reachability tests confirmed this, even though all routing configuration and diagnostic output showed no problems. Further debugging (packet captures) showed that packets were arriving correctly on the relevant LVS balancer hosts, but they didn't respond.

At this point I suspected Linux’s rp_filter, the Reverse Path forwarding filter. This is a sysctl setting that filters any traffic received from an interface that is not also configured as the destination outbound interface for this (source) IP - essentially making sure asymmetric routing is forbidden, for security reasons. This default setting is appropriate for most hosts (including servers), but not for Linux hosts configured as advanced routers with multiple interfaces, such as our LVS balancers. Therefore we explicitly disable this filter in configuration management. I quickly verified whether this configuration was still present in sysctl.d (it was), and also manually disabled rp_filter on one LVS balancer to be sure. It didn’t change anything. Unfortunately I had used sysctl net.conf.ipv4.all…

At this point I suspected there was some other configuration issue on the LVS balancers that I had rebooted. There was one remaining LVS balancer for public traffic (lvs1005) that hadn’t been rebooted yet, and 2 for internal traffic. Around 11:30 UTC, I manually copied relevant LVS service configuration to lvs1005, assigned some LVS service IPs to the host and configured a static route on the active router for internal traffic. This quick & dirty hack ensured that at least HTTPS traffic for wikis started to work again, while I could diagnose the underlying problem.

However, lacking any iptables firewalling on these hosts and all routing output looking correct, there was little else I could think of besides rp_filter, so I decided to investigate that again. I found that rp_filter was still enabled despite having manually disabled it before - or so I thought. When I disabled rp_filter for eth0 explicitly, immediately LVS service IP reachability for this balancer was restored. After I manually disabled rp_filter for eth0 for the other balancers, all problems were immediately resolved at 11:44 UTC.

The net effect of this configuration error was that LVS services were broken for _most_ hosts inside of our network after reboot of the respective balancer hosts, but working fine for all external traffic. This implies that most LVS services appeared to work fine from the outside, including any manual tests run across the Internet. But services that internally forward to other services, such as HTTPS (in eqiad), Parsoid and Swift didn’t work.

I traced the underlying problem back to this Puppet change: https://gerrit.wikimedia.org/r/#/c/75087/

This is a big puppet wide change that modularises and improves the handling of sysctl parameter values across our infrastructure. Our LVS balancers rely heavily on specific sysctl settings, including disabling rp_filter. Unfortunately, despite reviews by 3 different people on multiple patch sets, it hadn't been noticed that this patch failed to explicitly set the priority of (at least) the LVS specific sysctl settings. Before, the sysctl priority was set at 50, and now it was at the default of 10 - equal to the priority used by the default Ubuntu sysctl settings. Due to alphabetic naming order, the Ubuntu sysctl file runs last, and reenables rp_filter again.

I’ve now corrected this in: https://gerrit.wikimedia.org/r/#/c/110940/

As for sysctl net.conf.ipv4.all.rp_filter, the kernel uses the max() value for {all,interface}, which is not very intuitive. So to reliably turn it off, it needs to be set to 0 everywhere. I've run into this several times before in the past and I should know this, but unfortunately I didn’t think of it immediately, which extended this outage by some 10-15 minutes. :(

Conclusions

- Never trust sysctl.net.ipv4.conf.all to do what you think it does. It probably doesn’t.
- Be more careful with assuming fail overs on passive/standby machines succeeded correctly. When they are not in use such is hard to verify and monitoring may not catch everything. In such cases, maintenance on the active part /first/ may actually be better.
- Big/wide Puppet change sets are bad and risky, no matter how trivial they may seem. Even multiple reviews can fail to spot issues, and these should be implemented and tested step wise on critical systems.

Actionables

Most of the issues addressed (as of 2014-03-19), the rest of the things like are good to haves, not critical.

- Status: 🟩 **Done** - We should explicitly monitor some critical sysctl active values on systems.
  - "add Icinga checks for critically important sysctl params" - Ori
- Status: ⬜ **Declined** - LVS testing needs to include internal services testing, and simple TCP port connects may not tell the whole story.
  - RT 6812
- Status: 🟩 **Done** - Check remaining uses of sysctl::parameters and their priorities (Andrew Bogott has committed to handling this).
  - "Restore sysctl priorities." - Andrew Bogott
- Status: 🟩 **Done** - We need to reinvestigate the performance impact of ntpd on present day LVS (which was found detrimental on old kernels years ago), or find a solution for maintaining the clocks on these systems if it’s still a problem.
  - RT 6813

Category: Incident documentation

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.