Thinking outside the box

How to be an effective sysadmin under pressure

Steven McDonald
Anchor
2014-01-06

What is “the box”?

- A confined thinking space.
- Confined by the person doing the thinking.
- Inside the box live past experiences, common wisdom and other existing knowledge.
- Thinking inside the box isn't really thinking, it's reusing past experience.

Why think outside the box?

- Usually, thinking inside the box is quicker, easier and produces the same result.
- Most of the time, you **want** to confine your thinking to “the box”.
- This is why we write scripts to automate so much in-box thinking. Out-of-box thinking cannot be scripted.
- It's important to know **when** to leave the box.

Scenario 1

- You run a web server that is being DDoS'd by a large number of remote IP addresses with randomly generated (but mostly realistic) user agents.
- The customer for this service is getting antsy because they have an advertising campaign going on and don't want to lose traffic.

Scenario 1: Solution

- Write a script to watch the access log files and count the number of different user agents associated with each remote IP address.
- Based on that information, the script would block IP addresses over a set threshold of user agents in iptables.

Scenario 1: Analysis

- This solution is far from a perfect one, as it would have many false positives for large organisations with web proxies.
- However, restoring the website for 90% of the user base is preferable to it being usable for 0%.

Scenario 1: Conclusion

- Sometimes a perfect solution is not possible in the immediate term.
- Implementing a partial solution is better than continuing to hunt for a perfect one.

Scenario 2

- You are migrating hundreds of GB of static assets to a new host with faster disks in preparation for a large traffic spike.
- It becomes evident that you will not have all the files copied to the new host before the increased traffic hits.
- You can't be confident in the old host's ability to serve the expected level of traffic.

Scenario 2: Solution

- Configure the web server on the new host to first look for the file locally, and then proxy the request to the old host if it is not found.
- Cut traffic over to the new host.
- Continue copying files across in the background.

Scenario 2: Analysis

- This is a really, really ugly hack. This kind of solution would never be implemented in production in an ideal world.
- This solution adds a dependency on an additional web server, as well as being inefficient and not easily supportable.
- However, given the strict time constraint of the scenario, it is the least of three evils because it guarantees the best possible performance while still ensuring availability of all resources.

Scenario 2: Conclusion

- Sometimes, there is no good solution that satisfies every requirement.
- Sometimes requirements can be revised, as this can be a sign that too ambitious goals have been set.
- In cases where the requirements are immutable, a working bad solution is better than leaving the problem unsolved.

Scenario 3

- You manage a system that uses a shared NFS filesystem as a temporary holding area to pass files around.
- The NFS server is experiencing strange performance issues that will take time to diagnose and fix.
- Meanwhile, the critical nature of this data flow path means that the website depending on this system is down.

Scenario 3: Solution

- Set up a web server on the nodes that need to send data, serving files from a local directory.
- Work with the web developers to have the receiving nodes fetch files over HTTP instead of NFS, using Redis to store the list of URIs to process.

Scenario 3: Analysis

- Fixing NFS would have taken hours or days due to the nature of the problem (which is outside the scope of this talk).
- Bypassing NFS takes about 30 minutes while code is rewritten to speak HTTP.
- We get paid to keep the website up, not to manage an NFS server. Cutting out a malfunctioning component satisfies that goal.

Scenario 3: Conclusion

- It is far too easy to focus on fixing the root cause, instead of the problem the customer actually cares about.
- The problem needs to be clearly defined **before** it is solved, otherwise you are heading in the wrong direction before you begin.

Scenario 4: To be continued...

- Return to this lecture theatre at 13:20 on Wednesday for a more in-depth case study!

Summary

- Thinking outside the box is unnecessary to solve most common problems.
- I have shown one partial solution, one bad solution and one unconventional solution to uncommon problems. All of these solutions have one thing in common: they are **effective**.
- Uncommon problems call for extreme measures, especially when coupled with tight deadlines.
- In a word, be pragmatic in your solutions.

Closing thoughts

- Any and all advice given here can and should be disregarded at your discretion.
- Yes, that includes the above point. And this one.

Questions