After Action Report
26-27 June 2019 Outages

This page was intentionally left blank.

Summary

On June 10th, 2019, we were notified by Linode, one of our server providers, that our servers paralysis.tx.us.darkmyst.org (paralysis) and nebula.uk.eu.darkmyst.org (nebula) would be brought down for a hypervisor update. The admin and support teams acknowledged the notifications and proceeded to make the necessary arrangements. On Wednesday, June 26th, 2019, Paralysis went down for approximately twenty minutes then came back up successfully.

The server we were more concerned about during the upstream maintenance cycle was Nebula, due to the fact that our services package (Atheme) is hosted on there. We attempted to roll services over to a backup server in preparation for the outage, but during our first attempt ran into a database corruption issue. After thirty minutes, we were able to remove the corrupt section of the database, after which we tried again to move services over to the backup server, but failed again due to a dependency error. We then brought services back up on nebula and decided to have staff on call in the event systems did not come back up as intended after the update.

Our fears were realized when at 0130 UTC we were notified by our monitoring system that Nebula had not came back up from its maintenance. The admin team continued to keep a concerned eye on the situation until 0300 UTC when Linode's maintenance cycle was completed. At that point, a network emergency was declared by the Admin Coordinator and steps were taken to restore network services in one form or another. At 0545 UTC services were restored using our backup server with a 48 hour old database, which caused issues with Atheme not have proper user presence status and switching channel topics to out of date entries. At 1555 UTC, we shut down services on our backup server and brought them back up on Nebula thirty seconds later. The connection was successful and services was now operating with a six to eight hour old database. Thirty minutes later the incident was considered resolved and the network emergency canceled.

Failure #1: Database Corruption Error

Summary:

In preparation for the Nebula downtime, we attempted to do a rollover of services to emergency1.azure.va.us.darkmyst.org, one of our on-demand emergency/backup servers. During the rollover attempt, there was an issue with dependencies on emergency1. The attempt was abandoned, and an attempt to restart services on Nebula was made when on startup Atheme reported a database corruption error. After 30 minutes of investigating, the corruption was cleared out of the database and Atheme was successfully restarted.

Technical Details:

A snapshot of the database and services package was made on June 26th, 2019 at 0121 UTC in preparation for the rollover. The snapshot was then converted into a tarball and transferred over to emergency1, extracted, and basic integrity checks were made. At 1843 UTC, the Admin Coordinator updated Paralysis's configuration to trust the services on emergency1, then executed a configuration reload, which Charybdis succeeded at with no errors. At that time, the Admin Coordinator announced his intent in the staff channel to proceed with the rollover. He then set services in read only mode, dumped the database to disk, and shut down Atheme cleanly at 1901 UTC. He then attempted to start the services on emergency1 and ran into dependencies errors consistent with the fact that Atheme was compiled with dynamic libraries instead of static (this was unknown at the time). The Admin Coordinator then proceeded to abandon the attempt and restart Atheme on Nebula. Upon runtime, Atheme produced the following error:

```
[redacted for opsec]@nebula:~/atheme$ ./bin/atheme-services
[26/06/2019 15:08:36] atheme [redacted for opsec] is starting up...
[26/06/2019 15:08:36] module_locate_symbol(): nickserv/set_core is
not loaded.
[26/06/2019 15:08:36] module_load(): module [redacted for
opsec]/modules/nickserv/set_core is already loaded [at 0xa972c0]
[26/06/2019 15:08:36] module_load(): module [redacted for
opsec]/modules/groupserv/main is already loaded [at 0xa9d760]
[26/06/2019 15:08:36] opensex: grammar version is 1.
[26/06/2019 15:08:36] corestorage: data schema version is 12.
[26/06/2019 15:08:36] groupserv: opensex data schema version is 4.
[26/06/2019 15:08:36] db-read-time: needed int at file [redacted for
opsec]/etc/services.db line 15119 token 3
```

`[26/06/2019 15:08:36] db-read-time: exiting to avoid data loss`
At this time, the Admin Coordinator notified the Network Coordinator, Support Coordinator, and other staff members of the situation. We first investigated whether it could be a permission error but it wasn't. Upon further reread of the error message, the Admin Coordinator opened the database and went to the line in reference, which appeared to be a vhost/cloak entry for a non-existent account. As a test, he proceeded to remove the line, and then services came back up in a normal manner. It is unknown at this time what caused the corruption, but a ticket will be opened on the Atheme GitHub at some point in the near future regarding this.

Effect(s):

Services was down for approximately 30 minutes. No attacks were reported during this time, and no data was lost.

Corrective Actions Taken/Planned:

- Planned:
    - Nightly backups of the database on to an additional server and/or Azure.
- In Progress:
    - Fallback services will be deployed on Azure for rapid recovery.

Failure #1a: Services Downtime

Summary:

After network services were restored on Nebula, a second, unsuccessful attempt was made to restart them on emergency1. This is considered to be a sub-failure of #1.

Technical Details:

Once primary services were restored and confirmed to be in good health, a second attempt was made to relaunch them on emergency1. As mentioned in the previous failure, it was unknown that Atheme was built with dynamic libraries when copied over. Actions were taken to rectify this by giving the Atheme folder the same path on emergency1 as on Nebula. There was an additional library error that was unexpected and could not be corrected after investigation with ldd (a command line tool). The attempt was abandoned and services was restarted for a final time on Nebula.

Effect(s):

Services was down for less than five minutes, no attacks or data corruption were discovered during this outage.

Corrective Actions Taken/Planned:

Simply put, this failure should have never happened. The Admin Coordinator should have tested the services package on the testnet instead of on production. The following changes will be made to avoid unnecessary services downtime in the future.

- Dual Authorizations
    - Any major network reconfigurations that aren't considered an emergency will require two or more IRCops or senior coordination staff members to consent. They will then hold for five minutes, then reconfirm their intent to each other. A record will then be made in the staff wiki regarding what was done, when it was done, if it was an emergency, and who approved or disapproved.
    - If Atheme is taken down for any non-emergency reason, at least two senior staff members must agree.
        - Senior staff includes the following:
            - Network Coordinator
            - Support Coordinator
            - Deputy Support Coordinator
            - Admin Coordinator
            - Deputy Admin Coordinator
            - Additional Staff Approved By Network Coordn.
- Statically-Linked Atheme Builds
    - All future production builds of Atheme will be statically linked to avoid future portability errors.

Failure #2: Nebula Restart Failure

Summary:

At 0105 UTC on June 27th, 2019, the host Nebula lives on was brought down for hypervisor maintenance by Linode. According to Linode's internal logs, the server came back up at 0155 UTC, but as we found out later was lacking networking capability. With it seemingly still down when the maintenance window expired at 0300 UTC, a network emergency was declared and steps were taken to bring services back up in one form or another. Services were fully brought back up at 0545 UTC using an older database backup from ~48 hours before. At 1239 UTC, a

staff member with the ability to contact Linode came online and opened a support ticket. At 1415 UTC, connectivity was restored and the Charybdis server on Nebula rejoined the network. Atheme was then shut down on emergency1 at 1555 UTC and brought up on Nebula fifteen seconds later. The event lasted fifteen hours.

Technical Details:

At 0105 UTC, the Charybdis and Atheme instances on Nebula went down due to the host being rebooted for hypervisor upgrades. Moments later one of our IRCops rerouted the network flow to compensate. First, a bit of background info: typically Linode has their boxes back on after twenty to thirty minutes. Approximately 20 minutes after the server went down, senior Coordinators had their first notice that something was wrong when Azure sent out phone calls, text messages, and emails notifying us that it sensed no connectivity with Nebula on port 6667. The Admin Coordinator acknowledged the issue at 0146 and began monitoring the situation, both checking Azure and attempting to SSH in every fifteen to twenty minutes. At 0400, the Admin Coordinator sent out a text message notifying the Network Coordinator that Nebula was down and unresponsive. The Deputy Admin Coordinator mentioned that they may have had a copy of services on phoenix.nsw.au.darkmyst.org, and shortly after in PM the Admin and Deputy Admin Coordinators began taking steps to restore services. At 0433 UTC, an emergency instance of Atheme was up and running on emergency1. There were several issues with our first attempt:

- Read Only Mode:
    - Due to the fact that the database was 48 hours old, and to avoid any issues that might come up with have two databases for Atheme, it was initially decided to run services in read-only mode. This created several issues, chief among them locking channel topics into place due to ChanServ's unwillingness to update its stored topic (and restoring the old one) and NickServ's unwillingness to let people log on, as logged-on states are usually also persisted
- Server Trust Issues:
    - The only server that had trust for the new services package was Paralysis. This was due to the how trust works with Atheme. If the server uplink and downlink passwords match, along with the host, Atheme will accept the server, but it won't trust that ***it is*** services it unless the vhost that services presents itself with is what it thinks it should be. The vhost that we had our backup Atheme configured with initially was services.emergency1.darkmyst.org. Only Paralysis had this vhost listed as trusted, and therefore it was the only server that recognized the backup Atheme's authority.
        - It was unknown to us at the time that all servers had to trust the backup Atheme, we were under the impression that just the connecting server had to. This was corrected in future attempts by having Atheme identify itself as services.darkmyst.org instead.

    - Since only Paralysis trusted the emergency Atheme instance, the rest of the network continued acting like services were unavailable, and wouldn't grant it certain restricted access privileges and flags that it needed to operate.

Our second attempt was a bit more successful, Atheme was no longer operating in read-only mode, but was still having trust issues. At that point, we made one more change that allowed the network to trust the backup instance and services was fully restored. At 0649 the Network Coordinator acknowledged the outage with Nebula, and at 1239 UTC, staff came online that could investigate the problem. They opened a ticket with Linode support shortly after, and at 1413 they responded with a potential fix. Their solution worked and networking was fully restored. Nebula successfully completed its netjoin and netburst at 1415 UTC. At 1555 services on emergency1 were shut down, and then brought back up on Nebula approximately thirty seconds later. Nebula reported a successful netjoin and netburst with it, and services functionality was confirmed shortly after. At 1625, the Admin Coordinator declared the network emergency over, and emergency1 was spun down shortly after.

Effects:

Services was down from 0105 UTC to 1555 UTC, for almost 15 hours. This created major disruptions by making all *Servs' unavailable, and additionally opening the chances of an attack on the network without our AKILL defenses from OperServ, along with heavily crippling the ability for channel operators to defend their users, and keep protected nicknames guarded. This was a severe failure in our network, and a failure to keep the trust you put in us safe. We are deeply sorry, and will work hard by implementing new safeguards and policies to keep the network and you, our users, safe.

Additionally, the services database was out of date by fifteen hours, and even further complicated by the fact we had the emergency1 services running with an even older database.

Corrective Actions Taken/Planned:

- Alerting
    - Our alerting system worked flawlessly, but our response to it was delayed. This is mostly unavoidable, factoring in our highly international team. A future improvement will be reacting quicker to alerts from Azure, that way we can monitor the situation closer and wake up people if needed.
- Backup Systems
    - A backup services platform will be maintained on emergency1 in the event primary services on Nebula fails or a maintenance cycle is needed.
- Handover
    - In the event of future maintenance on Nebula, services will be handed over to emergency1 no more than two hours before the window goes into effect.

- Access
    - The Admin Coordinator now has access to the management panel for Nebula, and therefore can investigate issues and open support tickets as needed.