Community suggestions for minimal DR preparation

These are the building blocks; the very most basic things that anyone will need in order to recover from a disaster that impacts their campus network connectivity and presence, for whatever reason. Almost everyone will want to do more preparation than this, but we believe that everyone will need to at least do these things, and that this is a reasonable beginning.

DNS

Most campuses have a combination of on-campus and off-campus authoritative DNS servers (those that hold records for the institution's names and addresses), typically with an on-campus primary or master server, and multiple secondary or slave servers. The master server, which keeps the original records for the campus, may be configured by hand, or through scripts or a web interface to make it simpler for support staff to handle requests for adds and changes, or may be configured automatically through DHCP. The slave servers retrieve copies of the records from the master in order to answer queries.

The DNS contains records that map names to IP addresses, and corresponding records for IP addresses to names (called reverse or PTR records). These are generally somewhat less important, but can be critical for campus email servers in particular since the existence of a reverse record is sometimes used to evaluate whether email is legitimate or spam. Some servers are configured to only accept Secure Shell (SSH) connections from connecting IPs that map to a fully qualified name, and which in turn maps back to the original IP. As a result, if the campus loses reverse DNS, you may experience outbound connectivity problems.

It is important to note that viewed from the outside world all of the servers, master or slave, appear identical. Each answers with the same information, and an external system can (and will) choose one randomly when making a query. If a server is unreachable, the querier will eventually time out and try another, typically after five seconds. Although a single five second delay may be acceptable for external users, if more than one server becomes unavailable and that delay is multiplied, the user may conclude that the website or other resource is unavailable.

There are a number of other timeouts associated with DNS records; two that are especially important are the time-to-live (TTL) and the expire. The TTL sets the time for which a particular record is valid after it has been fetched. To avoid excess load on the servers, many records are set with a TTL of 86400 seconds, or one day. This means that once a remote server has fetched a copy of the record, it will be kept, or cached, for a day before it is requested again. While this reduces the load on the servers, it also means that any changes to the record will take up to a day to propagate to all remote systems. Any emergency changes, such as to point to a backup server, or deflect an incoming denial of service attack, will be significantly delayed.

On the other hand, a very small TTL like 300 seconds means that changes will propagate quickly, but also means that if all of the nameservers become unavailable for some reason the record will quickly disappear from the remote caches and the host that it represents will no longer be reachable.

The expire value is the amount of time that a slave server will keep a copy of the DNS zone (all of the names in a particular domain) and continue to use it to answer queries, if it has not had the opportunity to contact the master server and check for any updates. This timeout is designed to ensure that the system can withstand an outage of the master server, while preventing extremely old records from being used. Any value can be used, and it is typically recommended that the expire timer be set to at least several days.

The timers that are associated with the entire zone are set in a special record called the Start of Authority (SOA), at the beginning of the zone file. It can be examined by querying the nameserver; for example:

```
% dig \-t soa uoregon.edu

uoregon.edu.            86400   IN      SOA     localhost.uoregon.edu.
hostmaster.uoregon.edu. 2009120709 7200 900 605000 86400

Those fields are:

root name of the zone: uoregon.edu.
TTL:                   86400
class:                 IN (Internet)
name-server:           localhost.uoregon.edu.
email-address:         hostmaster@uoregon.edu (the first dot is imputed as being an @ sign)
serial-number:         2009120709 (common practice is to use a timestamp)
refresh:               7200 (time between slaves refreshing from the master name server in seconds)
retry:                 900 (time between retries if the slave fails to connect with the master when refreshing,
in seconds)
expiry:                605000 (time in seconds til a secondary copy of the data from the master is no longer
valid)
neg-cache-time:        86400 (time in seconds that a NEGATIVE (NXDOMAIN) answer should be cached)
```

In the typical model of a master server on campus and slaves both on and off-site, a failure that results in a loss of campus connectivity will cause several immediate effects:

- Queries will continue to be answered by the remaining servers using the last copy of the DNS zone, but they will not be able to be updated.
- There will be some delay for each query as the remote side attempts to contact servers that are now unreachable, varying based on the order in which the servers are queried and the number that are still available.
- Once the expire timer has run out, the zone will disappear.

Considerations

Some mechanism must be established to allow one of the off-site slave servers to become a master, and to reconfigure the other slave servers to request updates from it. This is an issue both for policy and procedures, and a technical concern. The DNS records on this server must be manually editable to allow for traffic to be redirected towards DR/BC resources, and to advertise a new list of nameservers, removing those that are no longer reachable. This can be as simple as a secondary set of configuration files with a copy of the DNS records (probably much reduced from the standard campus zone) and a procedure or script to restart the nameserver software with the new config. The new master server will then notify the slaves that they should fetch an updated set of records, and they will all begin serving the new information.

TTLs for any hosts that are part of the DR/BC system should be carefully considered, with an eye towards maintaining continuity of service in the event of a temporary DNS outage, but allowing for sufficiently rapid transition to the DR/BC servers when needed. With modern servers and high bandwidth connections, the added load from more-frequent queries is typically of little concern, and TTLs as short as 600 seconds may be appropriate. It is critical to balance the stability of the system in the event of a failure and the requirement for rapid changes when responding to the failure.

If the DR/BC servers are in a separate network address range, provisions should be made to provide reverse DNS services for those addresses, either through a network provider or (preferably) using the off-site campus-owned DNS servers. Remember to consider the campus reverse DNS as well when planning for DR DNS support.

In order to move zone files from the master server to the slaves, TCP connections from the slaves to port 53 on the master must be allowed. This may impact firewall or router access control list configuration.

Remote Access

In many disaster scenarios, system administrators will require access to the DR/BC equipment from off-campus locations, perhaps with unpredictable connectivity: home broadband connections, cellular modems, even public access wireless networks. Most campuses have VPN access or allow SSH connectivity to a gateway host, or even direct SSH to internal servers. This functionality needs to be duplicated for the DR/BC environment, with the added requirement that it must not depend on any on-campus resources like directory or authentication servers.

Considerations

SSH access may be sufficient for system administration, but some OSes and many administrative applications will require VPN access and remote desktop capabilities. Once again, the remote desktop login must be independent of campus authentication, either by being self-contained or by utilizing a DR authentication infrastructure as described below. Since some administrative apps exchange unencrypted traffic, VPN access is critical to avoid system compromise when users are accessing the application remotely during an emergency.

It is likely that additional remote access will be requested during the response, as well as user support requests like password resets, requiring both a procedure for filling the requests and a policy for granting authorization. This is an area where regular testing is especially helpful.

Although many sites have abandoned dialup modems as a regular network access method, they can provide an inexpensive and reliable route for emergency out-of-band access, particularly for system administrators to make remote configuration changes during an emergency.

Authentication Infrastructure

E-mail - sending and receiving

A typical university mail infrastructure is quite complex, including:

- server farm accepting incoming email
- inbound spam and antivirus filtering
- IMAP and POP servers providing user email access
- webmail servers
- shell login access
- specialized servers (Blackberry, etc.)
- name and address directory service (via LDAP or some other mechanism, potentially directly querying administrative servers)

There may also be multiple parallel mail infrastructures; for example, a Unix-based cluster for faculty and most staff, an Exchange-based environment for senior administrators, outsourced email for students or alumni, and so on.

Since the email environment is likely to be complex and heavily built, it is tempting to consider a cut-down version for emergency use that would remove certain components like spam filtering or restrict users to POP access rather than IMAP or webmail. This may be a false economy, however, if the unfiltered mail stream overwhelms users with spam (which in many cases is several times the volume of legitimate email) or requires the users to change their email reading process, reconfigure applications on laptops or home computers, etc.

Another factor arguing in favor of a complete mail backup system is the degree to which most institutional users regard email as a critical resource; a DR system could be pressed into service simply because of a failure of the main campus email infrastructure, during upgrades, or as a test and development platform to verify new code before deployment.

Considerations

If a complete duplicate of the email infrastructure is not practical, an obvious place to start is the system that provides for the senior administrative and support staff. Although this is likely to be suitable only for very short-term use, the availability of a stable email server that allows users to communicate with their existing, well-known email addresses is critical during an emergency.

If users are allowed to store email on the campus servers, there may be synchronization issues during switchovers between the primary and DR systems. This is particularly an issue when moving off of the DR infrastructure back to the primary servers.

Backup systems that rely on users' ability to reconfigure their email clients or even to change email reading behavior will have dramatically increased training, testing and support requirements.

Email will also be a critical outbound communication path during an emergency. Sudden changes in email sending behavior may trigger anti-spam measures, so the network provider for the DR servers should be made aware that there can be significant volumes of outbound email during an emergency.

Website

This system is where many DR plans begin; the need for a backup web presence for the institution. Given the importance of the campus website for internal and external users, it is a critical consideration for the DR setup. However, without the other systems listed above it will not be possible to activate, use and manage a backup website.

Of course, just as email is no longer a simple matter of one standalone server, so, too, the campus web site is probably no longer a single server. A typical university web site might have:

- a farm of web servers, sitting behind a
- load balancer, fed by a
- content management system, and including
- data driven content fed by MySQL, Postgres or some other database, with its own servers

As with the email infrastructure, although a complete hot backup website system will be complex and expensive, it can provide security for this critical service during on-campus outages, upgrades, etc.

Considerations

A basic, static web site with status information and provisions for remote updates is extremely simple to configure and maintain, and should be the lowest common denominator configuration for every site. In some cases it will be possible to add complexity incrementally, as time and resources permit.

During an emergency that involves threats to the members of the university, public demand for information will invariably lead to dramatic increases in web traffic. The server(s), network equipment and external connections at the DR site must be sized to allow for such surges. A status-only page, or a low-impact version of the campus page, can reduce the impact of a sudden flurry of requests.