Operational Testing of Your Disaster Recovery Plan

PivotPoint SECURITY

Where to turn.

How well do you really know your disaster recovery plan? Chances are you really do not know if your recovery plan is going to work until you test it. No matter how good you are at determining requirements and developing plans, no one gets everything 100% right coming out of the gate. Interdependencies, data flows… if you do map everything correctly the first time, take that extended vacation—you have earned it.

For example, Marketing might say, "We are strategic, so if we are down for a week it is not going to hurt anybody." Then you talk to Sales and they say: "We have to be up within two days, and to do what we do we must have this data feed from Marketing, without which we are dead."

Interdependencies between systems often become so matter-of-fact, accepted, misunderstood or just plain invisible that very few people really know where all the information comes from that populates the systems they rely on to do their jobs. So unless you test your recovery plan, you are not going to know what you do not know.

Interdependencies between systems often become matter-of-fact, accepted, misunderstood or just plain invisible...

Unfortunately, most recovery plan testing is limited to a tabletop—not an operational failover. In a tabletop test, you pull out your recovery plan, review it, and talk through a scenario.

At that level, everything may look fine. But it is only through a failover that you will shake out the bugs and ensure that your alternate facility or alternate processing capability accounts for all the actual interdependencies.

Upgrades are another area that could cause recovery-related issues. When Marketing switched from System F to System G, was the recovery plan updated with all the infrastructure changes? Unless your documentation is current and ports over to your recovery plan, you may be missing connections that were in place previously. That can have a significant impact on recovery, slowing down the entire process as delays trickle down to other systems that may need technical attention.

Another question is: Are the backup systems (still) configured the same way as the primary systems? Unless you are operationally testing them, you do not really know. Some organizations have found out the hard way that their backup systems were not compatible with their production systems, because the primary systems were upgraded to, say, Oracle 12c but the backup systems were still running on Oracle 10g.

The fact is that recovery/business continuity testing must be done operationally so that if something goes wrong, it goes wrong in a safe environment where your business and its reputation are not at stake and you do not have to put your neck on the chopping block. Then afterwards you can go about your business with a much higher expectation of recovery success and a more accurate understanding of your true recovery capability.

Here is another thing an operational test will validate: Say you have 100 systems in your data center. The 20 most critical of them need to be recovered within 12 hours, while the least critical 20 can be delayed for 10 days or more, and the rest fall somewhere in between.

Say you estimate that your techs can recover each critical system within four hours. But you have 20 systems that have to be recovered within 12 hours. Do the math: If you can recover one system every four hours that means in 12 hours you can bring up 5 systems, if you work around the clock. Unless you conduct the operational test, it is just that—an estimate. Responding to an actual disaster is not the time to find out your estimates were off.

So what do you do then? Either adjust your recovery requirements, choose a different strategy that allows you to bring systems up faster, or hire more IT staff.

Until you conduct operational failover testing, you are never really sure you can recover your systems within recovery time objectives (RTOs) and recover your data within recovery point objectives (RPOs). This is how you learn things like: We can bring up System 1 in three hours, System 2 takes only two hours, and System 3 takes eight hours.

Tabletop testing is great, and it has its place, but it cannot give you the real, complete picture of whether you can fulfill your recovery commitments with the assets and strategies you have in place.

Any exercise is not something you do off-the-cuff. There are several different approaches you can take, but they all require forethought and preparation.

In between an operational recovery test and a tabletop is a walk-through—rather like a "tabletop plus" because a walk-through also includes an inventory. As you talk through your response to a hypothetical incident, you perform an "eyeball" inventory: Are the fire extinguishers in place and inspected? Are the backup tapes in the bin where they are supposed to be? Are the relevant SOPs on the SharePoint site in the folder where you think they are, and are they current?

In terms of actual operational tests, there are two flavors:

2

1. **Parallel processing**, where you continue to operate using your primary systems as usual, while a select subset of users access the backup systems and make sure they can still get work done; and
2. **A simulation**, where you actually turn off your primary systems, initiate your recovery plan and truly see what works and what does not.

Simulations take time, training, correct documentation, current backups… and if you do not have a mature recovery capability, something may very well go is wrong, especially the first time through, which could very well be disruptive to normal operations. But that is the nature of disasters—they are disruptive to normal operations.

Many organizations are understandably hesitant to perform a simulation. But that mindset creates a Catch-22: You do not think your recovery capability will stand up to the testing so you do not test it. Thus, you do not know what you do not know and when a real disaster strikes your recovery may not be as efficient and effective as you need it to be. As I rhetorically asked above: Do you want to find this out in a controlled, simulated environment, or in an actual disaster?

A simulation is the foundation of continuous improvement in your recovery capability.

A simulation is the foundation of continuous improvement in your recovery capability because it really shows you what works and what does not. The first test-drive may be bumpy but then you will have a wealth of recovery-related considerations and experience that you can incorporate into everyday operations to provide you with a far more robust recovery capability.

If you run simulations regularly and incorporate lessons learned into everyday operations, you can achieve resiliency. Resiliency is the recovery end-state where everyone is so comfortable with how and what to do if an event occurs that there is no guesswork. Critical functions continue within their RTOs and customers may not even notice you had an issue.

Having said all of that, the question remains: How do you develop a realistic and worthwhile exercise scenario? There is considerable homework involved. Start by taking a look at your risk assessment to see what disruptions are most likely to occur, and pick one. You might want to consider designing a scenario that will enable you to exercise your incident response (IR) capability and then invoke your recovery capability based on those results.

If your industry is getting slammed by viruses and data breaches, you might pick a cyberattack or malware infection as the cause of the outage for your exercise.

For example, start with the help desk. Tell them there is a simulated system failure. They should refer to their procedures (which you read when you wrote the scenario so you know what they would do) and escalate the problem to the Incident Response (IR) team. When the IR team comes up with an estimated time to repair (ETR) that exceeds the RTO for the impacted system(s) (which it will because you figured that out when you wrote the scenario), the IR team should coordinate with the business continuity/disaster recovery folks, kicking off your BC/DR plan.

Here is another scenario example: You have a fire, which kicks off your emergency response protocol. How do you get people out the door, respond to things like evacuation and building shutdown, etc.? Then have the impact of the incident result in outages that exceed RTOs for key systems, which should cause BC/DR plan implementation.

But since your scenario will be simulated, how do you move it forward? You develop what I call *injects* (the U.S. federal government calls them the Master Scenario Events List (MSELS, pronounced “measles”… really). Regardless of what you call it, these are the tidbits of information that allow an exercise to move forward.

You might walk into the help desk and hand them a scrap of paper that says: “Five users called in in the last ten minutes saying they cannot access System X.” Then the help desk goes through its procedures and does its assessment. Then a few minutes later you hand the help desk another scrap of paper that says: “Investigation determines that the system is down” or the database is corrupted, or whatever it is you are going to put them through.

DR – **Disaster Recovery**

BC – **Business Continuity**

RTO – **Recovery Time Objective**

IR – **Incident Response**

ETR – **Estimated Time to Repair**

MSELS – **Master Scenario Events List**

In other words, you ratchet up the activity based on a series of carefully planned, hypothetical injects that tell people what they would find if they were working from procedures and policies to respond to an actual disaster.

What if you want to complete your exercise in four hours and your recovery capabilities do not kick in until twelve hours into an event? Using injects, you develop your exercise timeline and map it to real-time. It is all part of how you design the exercise: How much “real time” do you want to simulate? How do you “fake” time passing to get responses to kick in as they would in real life?

Other things to consider when developing a scenario include: Who needs to play, and where are they going to play from? What systems are involved, etc.? Are we going to serve coffee and donuts? (People are more likely to show up for exercises if you feed them.)

Among the key players in your scenario are your BC coordinator and possibly the command team. You will also need someone to act as a facilitator. This is the person who hands out the injects and tracks the exercise, usually your BC coordinator. Working with the facilitator will be monitors. These are the people walking around observing and evaluating the responses. For example, if IT is involved in your scenario, the IT director might be walking around monitoring the IT staff's responses.

Regardless of who is involved, at the end of the day, everybody should be providing feedback: What worked, what did not work, what procedures were not complete, what documentation was not available, what would have been "nice to have"—all that stuff. Based on this feedback and experience you then update your IR procedures and DR plan.

Then when a real incident occurs and a real disaster is declared, you will be ready.

Everybody should be providing feedback:

- ☑ **What worked, what did not work?**
- ☑ **What procedures were not complete?**
- ☑ **What documentation was not available?**
- ☑ **What would have been "nice to have"?**

To discuss Disaster Recovery and Business Continuity Planning (ISO 22301),
Contact Pivot Point Security.

Research compiled and written by Robert Cohen, Certified ISO 22301 Lead Implementer, Certified Business Continuity Professional (CBCP and Certified ISO-27001 Lead Implementer. Bob joined Pivot Point Security in 2014, having worked for over 22 years in DRBCP and Information Security consulting, including disaster recovery and Continuity of Operations (COOP)implementation and training for government agencies, such as, NOAA, NASA, Department of Transportation, Department for Health and Human Services, and Department of Defense and Fortune 500 businesses. Bob has been published several times, most notably in Security Volume (Volume II) of the HIPAA Implementation Guide. Since joining Pivot Point Security, he has focused primarily on ISO 22301, business continuity and audit/ISO implementation projects.