




The impact of domain name collisions: a business perspective

Mike Sax

Chairman and Founder, Association for Competitive Technology

ACT | The App Association
1401 K Street NW Suite 502
Washington, DC 20005

 202.331.2130

 @ACTonline

 ACTonline.org

 /actonline

Contents

Introduction.....	3
Internal networks are critical.....	3
Projected disruption	4
Effects of a domain collision	5
A day in the life of a domain collision	7
Crippling for SMEs	8
Not an easy fix	9
Additional challenges with custom software.....	10
The internal domain renaming procedure.....	10
<i>Compiling a comprehensive inventory.....</i>	<i>11</i>
<i>Create a dependency map.....</i>	<i>11</i>
<i>Develop a phased migration plan.....</i>	<i>12</i>
<i>Update custom software.....</i>	<i>13</i>
<i>Simulation in a test environment</i>	<i>13</i>
<i>Execution and verification.....</i>	<i>13</i>
Pointing fingers is counterproductive.....	14
What can be done?.....	14
<i>Raising awareness.....</i>	<i>14</i>
<i>Real world testing to aid risk mitigation</i>	<i>15</i>
<i>Stay Informed.....</i>	<i>15</i>

Introduction

Stakeholders within ICANN have looked at the problem of domain collisions using an outside perspective, studying the number of businesses that may be affected by the problem and which processes can be put in place to reduce those numbers.

In this paper we explore this problem using a different but equally important perspective: we examine how exactly small and large businesses are impacted when they fall victim to a domain collision. We look at how these problems will emerge, how they will impact the daily operations of the business, and what IT departments need to do to address the issues.

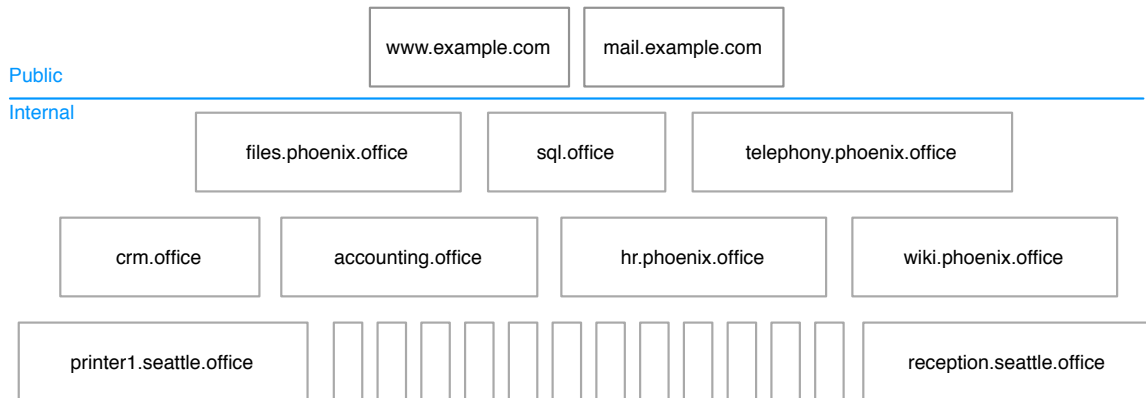
Internal networks are critical

While the Internet has an enormous impact on e-commerce and the interaction between businesses and with consumers, this is really only the tip of the information technology iceberg. The vast majority of the world's IT is crucial to support the internal operations of businesses. Internal communications, planning, accounting, human resources, and manufacturing software all run on internal servers. These servers provide essential services to other servers as well as client devices such as computers, tablets, and mobile phones.

The IT infrastructure of a business is a web of interdependent clients and servers. Applications depend on storage and database servers, printers, storage devices, etc. Some of these applications were purchased from third party vendors; others may have been developed internally. All of them need a way to identify other servers on the network.

Just like on the global Internet, every system on local networks has a name. These names usually use a multi-level structure similar to public Internet domains. For example, a national organization with multiple locations could define a local domain for each city where it has an office, such as `seattle.office`, `phoenix.office`, and `denver.office`. Under each of these local domains there may be other systems such as

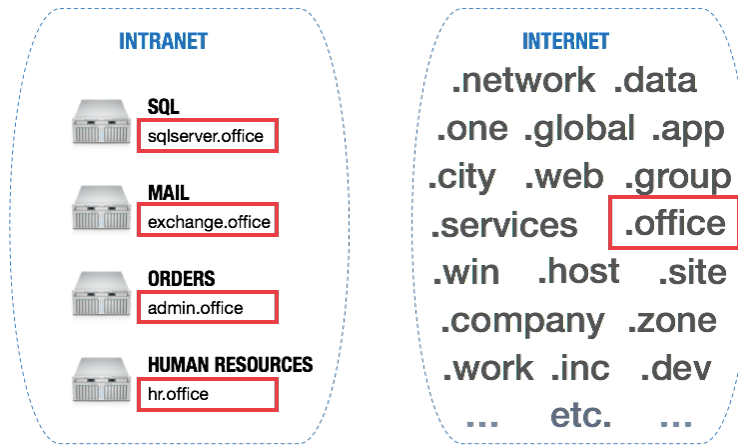
mail.seattle.office, crm.seattle.office, etc. The network has its own set of internal DNS servers that resolve internal domain names to the correct systems within the organization.



Each company may have a different naming structure, choosing one or more (local) top-level domains for their internal network. Some may have decided to use .office, .network, .global, or any other top level domain that was not in use at the time. When their local network was first created, it was not obvious that any of these top-level domains would one day be up for auction.

Projected disruption

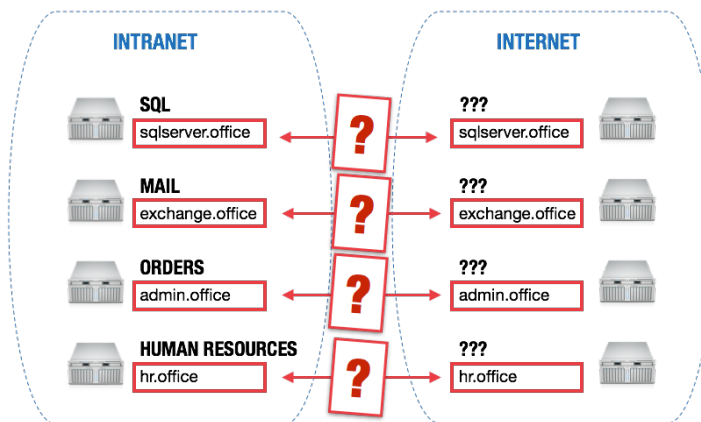
As more than 1,400 new top level domains will be introduced during the coming months, there is a considerable likelihood that that these will overlap (or collide) with the existing local top level domain names that companies have selected to manage their own internal networks. When this happens, chaos will ensue: the entire DNS system is based on the principle that a single domain name resolves to one system or at least one group of systems that behave in a uniform and consistent fashion.



When a single domain name can resolve to either an internal computer or a system that is under the control of a complete stranger, it may cause massive failures across the entire network and leave the door open to unacceptable security breaches.

Effects of a domain collision

What exactly will happen when a company’s internal domain collides with a new top level domain? What are the effects of ambiguous domain names that may resolve to either an internal system or an external public server?



DNS servers will commonly first look for internally defined names, and only query external DNS servers when the internal name is not found. In this case, systems that

are connected using the internal wired network would continue to function as before.

Unfortunately, older versions of Windows from as recently as a few years ago are configured by default to query DNS externally first, and only internally when the external DNS server could not resolve the name. When a domain becomes fully registered on the Internet, these networks will break down immediately because the servers needed to run internal operations can no longer be found.

For systems that resolve names by checking internally first, the problems will be more intermittent. Computers connected to the wired internal network will most likely continue to resolve names as expected. But what if the user is using a laptop or mobile device? Which system, internal or external, will these domains resolve to? The short answer is: it depends.

If the user's system is connected using a public wireless network, for example at a local coffee shop, public DNS servers will resolve names to the public servers. If users connect to mail.seattle.office and see a page that looks like the login page of their regular mail server software, they will not suspect anything.

Users will enter their credentials without hesitation, even though they are most likely connected to a server that is under the control of a complete stranger. If they are lucky, this will simply be another company's mail server and they won't be able to log in. But it's also possible that the system is under the control of a malicious agent who registered the domain with the sole purpose of tricking people into giving away their credentials.

If users connect using a Virtual Private Network (VPN), they are more likely to hit the internal domain servers and resolve to internal systems. However, there is also the problem of internal domain caching. When access is attempted to certain servers before the user actually logged into the VPN, the IP addresses corresponding to these domains may be cached for several hours. In that case, an originally public

server address returned before logging into the VPN will continue to cause havoc and expose major security risks. Worse, the problems will appear intermittently, with no apparent cause.

A day in the life of a domain collision

Consider what would happen when documents.office suddenly became a public domain. We imagine a small law firm that has a document server set up using the name documents.office:

Day 1,
7:45 AM Jennifer checks a few things at a local coffee shop before going into work. When she opens her laptop, an application checks documents.office. The laptop is not connected to the VPN yet and the coffee shop DNS server returns the IP address of an external server that was registered on the Internet as documents.office.

Day 1,
8:03 AM Arriving at her desk, Jennifer puts her laptop in its docking station and gets to work. When trying to open the document she was working on the day before, she gets a file system error.

What happened was that her laptop was caching the DNS entry for documents.office from when she opened up her laptop at the coffee shop.

Frustrated, Jennifer waits a little bit and focuses on another task, hoping that the problem will go away.

Day 1,
8:50AM The document is still unavailable. It needs to get do the client by noon, so Jennifer is getting a bit anxious. She calls the consultant who handles their IT needs and they spend about 20 minutes troubleshooting on the phone. The problem appears persistent, so the

consultant agrees come by as soon as possible.

Day 1,
9:50AM Jennifer has been waiting patiently for the IT consultant to arrive. He's finally here, and asks her to show him the problem. She opens up her word processor and... it works! The consultant leaves and Jennifer gets to work.

What happened? The DNS cache, set for two hours for this domain, expired at 9:45AM, exactly two hours after Jennifer opened her laptop at the coffee shop.

Day 2, etc. Unfortunately, this is not the only occurrence of this failure. In the following days, at seemingly random times Jennifer is still unable to access documents. Jennifer and her coworkers are frustrated. The IT consultant has made numerous trips and is puzzled by these random failures. Eventually the cause of the problem (the domain collision) is identified and the law firm now has to pay significant amounts in service fees to reconfigure all devices to use a different top-level domain for its internal network infrastructure.

Crippling for SMEs

These intermittent failures can be especially troubling for small businesses. Scott Golightly, an IT consultant specializing in small business, explains how the impact of a domain collision can be devastating:

"In small companies with no IT department, essential tasks will start failing randomly and nothing anywhere is telling them why. It suddenly fails, they don't know why. They'll need to hire expensive consultants to investigate, while suffering devastating down-time."

Not an easy fix

Once a domain collision has been identified as problematic, how can it be fixed? Even when companies receive ample warning of the domain collision issue, implementing a fix can be a very time consuming and error-prone process. For example, Microsoft's "Step-by-Step Guide to Implementing Domain Rename" is an 87 page document filled with detailed instructions, warnings, worksheets, and recommendations. From the introduction:

It is extremely important and highly recommended that you test the domain rename procedure prior to performing it in a production environment. First, perform the domain rename procedure described in this document in a test environment that has a minimum of two domains.

Because domain rename is a complex procedure that affects every domain controller in your forest, you will find it very helpful to achieve headless management of the domain controllers in your forest before performing the domain rename procedure.

Familiarizing yourself with the specifics of each step in the domain rename process in a test environment will provide you with not only a much better understanding of the procedure itself, but will also better prepare you to troubleshoot any issues that arise during execution of the procedure in a production forest.

In addition the network systems software, all applications and services running on the network will need to be reconfigured. Because the change is so massive and penetrates deeply into a web of interdependencies, there is really no way to fully prepare for this in advance. Even with the most meticulous preparation, significant down time is to be expected and it is likely that there will be a long period of small fixes and lingering failures.

Additional challenges with custom software

Internal operations often depend on custom developed software. Mark Dunn, an enterprise software consultant who specializes in training teams of corporate developers, explains the core challenges related to custom software:

Enterprise software was often developed 15 to 20 years ago. Many applications have evolved into millions of lines of mission-critical code. Server name dependencies in a configuration file or library aren't examined in simple refactoring. At that point, bugs are introduced that can be extremely time consuming to find and fix.

When many of these applications were written, nobody imagined that it would ever become necessary to change all the server names that the application depends on. In many cases, the original authors of the software have already retired or moved on to different companies.

In addition, many of the developer tools and libraries that older internal software depends on are no longer sold or supported: PowerBuilder hasn't been sold or supported since December 31, 2011, and Visual Basic 6 was discontinued and is no longer supported since April 8, 2008. Software components used to build these applications are often no longer sold or their manufacturers are no longer in business. Reconstructing the development environment and simply recompiling the applications can be very difficult or impossible.

The internal domain renaming procedure

If a company learns that its internal domains need to be renamed, careful planning and flawless execution are required to minimize downtime of the company's operations. Even the slightest mistake may cause critical systems to malfunction. The domain renaming procedure consists of several phases:

Compiling a comprehensive inventory

The first step is to compile a comprehensive inventory of all systems and software that depend on the domain names in question:

- Network routers may have a managed interface or need to be reconfigured manually.
- Telephony servers and client devices (IP telephones) may need to be reconfigured. For some systems, this configuration may be very tedious and cumbersome, requiring additional time to be set aside for reconfiguration.
- Systems reconfiguration on application servers, desktop computers, and mobile devices is necessary. If these devices are remotely managed they may be able to be reconfigured *provided that the remote management software does not depend on the domain names that need to be renamed*. Otherwise, a very labor intensive and error prone manual process will need to be planned.
- Software configuration for each application running on servers, PCs, and mobile devices needs to be updated. In many cases, applications depend on specific servers or file locations for storage and database access. There is no uniform way for this type of configuration to be stored, so a detailed inventory is necessary detailing the steps required to modify settings.

Settings may be stored in files, in the registry, or in an internal format that requires manual intervention using the application's user interface. Some applications support centrally managed configuration, which greatly reduces the amount of work involved. Internally developed applications that have server names embedded in the code need to be identified. This may require a careful code analysis of all internally developed software.

Create a dependency map

Based on the inventory, it's important to create a map of dependencies between different software systems. This helps determine how systems will be affected by a name change. Based on this information the IT staff can determine the order in which changes should be implemented.

In addition to the dependencies, the map should mark the importance of each system for the continued operations of the business. For example, the correct operation of a simple device such as a printer that creates shipping labels may determine whether orders can go out the door. The computer that controls the printer may depend on a database server or a file location that is serviced by an affected system.

Although the complexity of a dependency map may quickly escalate to the point where everything appears to depend on everything else, identifying critical systems is essential to determine priorities in migration and testing.

Develop a phased migration plan

A migration plan will take into account the priorities and dependencies discovered in the inventory and mapping process. For each type of device or software, the plan will outline the exact steps required for configuration, preferably through systems management software but often through manual intervention. By including a time estimate for each step, IT staff can estimate the total downtime and impact on the company's operations.

Before any renaming can take place, it's highly recommend to perform a comprehensive backup of all systems involved, including routers, devices, and computer systems. Many systems cannot be fully backed up in an efficient, automated way, so the IT staff will need to make trade-offs between the time involved vs. how critical systems are for the continued operations of the organization. As a result, the "point of no return" may be reached fairly early during the migration process.

The migration plan may require several cycles of shutting down a number of systems within the organization. Because of the far-reaching nature of domain name changes, the whole plan will need to be executed as an uninterrupted operation, which may be problematic for organizations that don't have regular allowed periods of downtime such as nights or weekends.

Because many of the systems and applications may require manual re-configuration, sufficient people power needs to be allocated to perform the actual work. If the company uses IP telephony, additional contingency plans should be made so users can at least reach the help desk when they have problems. Reconfiguration often requires special administrator-level authorization. Careful planning is required to give people sufficient privileges to perform the necessary work without additional security risks.

Update custom software

Internally developed software may have server names embedded in the code. This software will need to be adapted to use a more dynamic method of configuration. For some older software it may be challenging to set up the build environment and simply get to the point where the existing code can be recompiled.

Simulation in a test environment

As much as possible, the migration process should be tested in advance. This may include setting up a test network environment with systems that include backups of existing servers and clients. While no simulation can ever replicate the complexity of the real world environment, catching problems early is much less expensive and disruptive than the alternative. Furthermore, a comprehensive test cycle will not only result in a deeper understanding of the procedure itself, but will also help in troubleshooting issues that appear during the execution phase.

Execution and verification

When a sufficiently large maintenance window has been scheduled, the different phases of the migration plan can be executed. Each step may require several hours to both execute and verify, and some phases may require lots of people power. Because the migration process will be executed outside of regular business hours, lots of overtime will need to be allocated. After the plan is complete and has been fully executed, the organization's IT staff will likely be busy for several days diagnosing and fixing unforeseen problems.

Pointing fingers is counterproductive

Was it foolish for these companies to start managing their network using top level domains that were not guaranteed to be perpetually unique? When a domain collision brings down their internal operations, will these companies simply get what they deserve? Miguel Castro, an IT expert who helps companies implement best practices has a more forgiving perspective:

“What can only be compared to the Y2K bug, some older systems may host names embedded directly in their code. The right-or-wrong aspect of this is irrelevant. Software design and techniques evolve over time and what we deem as incorrect and a poor design today, might have had a good reason back when it was written. “

The comparison with Y2K is very pertinent: the problem is widespread and affects all kinds of businesses, large and small. Without corrective action, the outcome is certain to be problematic and potentially catastrophic. Although Y2K ended up causing only very limited harm, the level of preparation and investment to address the issues in advance made an enormous contribution to limiting the damage. When looking at domain collisions, it is essential to keep in mind that the risks can be significantly mitigated with appropriate education and preventive measures.

What can be done?

The threat of domain collisions is real and substantial for thousands of businesses of all sizes. Businesses can easily find out if their local networks are at risk, and with sufficient preparation, are able to mitigate the risks and reduce the impact of such collisions.

Raising awareness

The difference between being able to plan a domain renaming procedure months in advance and being caught off guard is enormous. Renaming a domain requires not only careful planning but also extensive scheduled downtime. Without preparation,

the impact of a domain collision can be disastrous. In order to preserve trust in the Internet, it is crucial that businesses that may be affected by domain collisions receive ample warning and sufficient information on mitigation procedures.

The ICANN community has the duty to raise awareness of the threat of domain name collisions. Without awareness, nothing can be done. Implementing a solution requires extensive planning, coordination, and testing. For businesses that are caught off guard, the business losses suffered by a domain name collision can be devastating.

Real world testing to aid risk mitigation

Additional testing needs to be done with actual domains to assess the impact and build templates for remediation plans. These templates may be used by businesses as best practices to identify areas that may be affected and implement robust remedies.

Stay Informed

ICANN has created a "command center" page with pointers to all news and information related to domain collisions:

<http://www.icann.org/en/help/name-collision>