Carve Systems
38 E Ridgewood Ave. #110
Ridgewood, NJ 07450-3808
P: (201) 632-3422
E: Info@carvesystems.com

Carve Systems

Dotless Domain Name Security and Stability Study

July 29, 2013

Prepared by:
Mike Zusman, Jeremy Allen, Rajendra Umadas

Table of Contents

Introduction

In May 2013, ICANN contracted Carve Systems LLC ("Carve") to perform a study on the stability and security implications of the proposed dotless domain name functionality. Top Level Domains are those most Internet users are familiar with (com, net, info, edu, gov, org, etc.). These domains form the foundation of the Domain Name System (DNS). All domains must ultimately have, or be, a Top Level Domain (TLD).

Presently, if an Internet user attempts to resolve a gTLD name, such as 'example', without any "periods", no corresponding Internet address is returned. If 'example' were permitted to be a dotless domain name, it would resolve to an Internet address. In practice, this would mean that a dotless site, such as http://example, could now be a legitimate destination on the Internet if dotless domain names are allowed.

With ICANN's new gTLD program there will be more gTLDs operated pursuant to contracts with ICANN. Some of the organizations that have submitted applications to manage new gTLDs have expressed interest in operating as 'dotless' domain names. The introduction of dotless gTLDs could have implications for the stability and security of Internet infrastructure.

The consideration of publicly resolvable dotless domain names has created concern due to the wide reaching scope of TLDs. The SAC 053[1] report from the ICANN Security and Stability Advisory Committee (SSAC) has detailed some of these concerns. A primary finding of SAC 053 is that Microsoft Internet Explorer, by default, places dotless domain names in its trusted Intranet Security zone. The report also raised further concerns. As a result, ICANN contracted a deeper study into the risks imposed by dotless domain names to the security and stability of the DNS system, and Internet, as a whole.

This report details the approach, methodology, and results of this study.

[1] http://www.icann.org/en/groups/ssac/documents/sac-053-en.pdf

Executive Summary

The study performed by Carve focused on the potential security and stability impact on the Internet if dotless domains became widely adopted. This section briefly reviews Carve's methodology and then highlights the results of the study.

The engagement was conducted in three distinct phases between 24 May 2013 and 26 July 2013.

1. **Methodology Design** – a custom methodology was created based on Carve's standard Technology Risk Assessment methodology, and was submitted to ICANN for comment and approval
2. **Risk Assessment** – as per the approved methodology, components of Internet architecture that could be impacted by dotless names were enumerated, and individual risks & test cases for these components were documented & designed
3. **Testing** – tests were carried out to determine if dotless domains have any stability or security impact on Internet architecture components identified during the Risk Assessment

Brief Methodology Overview

The methodology was based on Carve's standard Technology Risk Assessment methodology, and focused on the people, process, and technology that would enable dotless domain functionality on the public Internet. Carve focused on a breadth of systems to ensure it surveyed as many pieces of critical Internet technology as possible within the time constraints of the study. For the detailed breakdown of the risk assessment, please see the Risk Assessment Methodology portion of this document.

Assumptions

Carve had to make certain assumptions when interpreting the results of the risk assessment regarding the impact of dotless domain names. Carve treats these assumptions as general principles of how software currently behaves and is reflected throughout modern technology stacks:

- A user of an Internet system needs to know where their data is being sent
- A user of an Internet system needs to know if a domain is internal (higher trust) or external (lower trust)
- Processes must account for namespace collision when provisioning internal names (individual organization processes), or allowing gTLDs (ICANN domain approval processes)
- Technology must correctly route data, and allow users to make informed trust decisions

Risk Assessment Results Overview

Carve identified twenty-five (25) individual risks in relation to the deployment of dotless domain names. Of these, ten (10) risks were considered to be of interest based on the systems they were related to, the damage they could potentially cause, or the amount of users they may potentially affect. All 25 risks are detailed in the Risk and Mitigation Table within Appendix A: Risk Listings and Potential Mitigations

General Observations

As a result of this study, Carve has identified **three categories** of concern that should be considered when analyzing the impact of dotless names on the stability and security of the Internet.

Namespace Collision Concern

A namespace collision occurs when a dotless name used on a private network becomes a resolvable name on the public Internet. The study confirmed that if systems are configured to use dotless domain names to locate intranet hosts, and these systems were to mistakenly use a public DNS server for name resolution, any dotless name collisions would cause the system to attempt to interact with the Internet-facing host. The study also suggests that users who are accustomed to accessing intranet resources via dotless names may unknowingly access untrusted Internet resources that share the same dotless names.

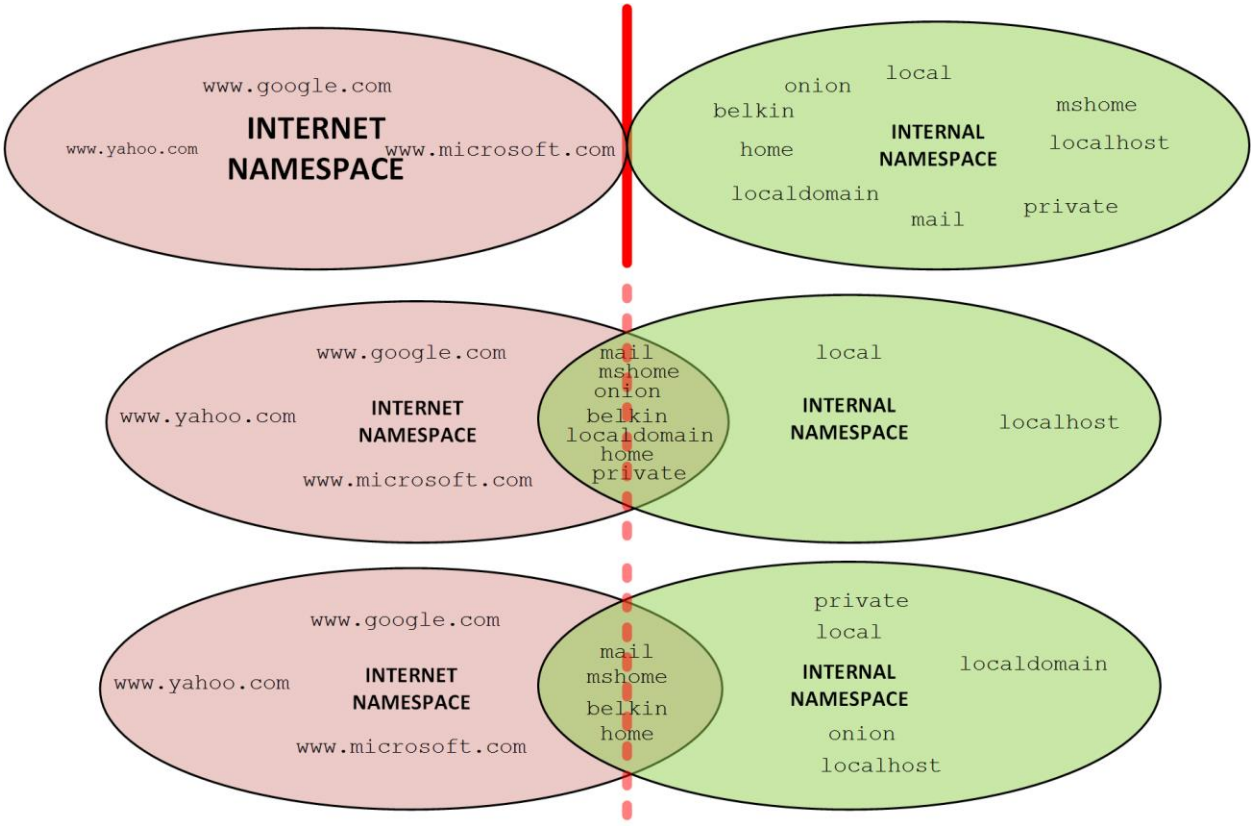

Figure 1: Namespace collision, from top to bottom: present, all dotless allowed, and 'managed'

User Confusion Concern

The second concern, **user confusion**, highlights the fact that dotless domain names have been primarily used on private networks for decades. This paradigm has created an expectation of trust, held by users and technology implementers, that dotless domain names always point to internal hosts, as opposed to Internet hosts.

Technology Confusion

The third concern, **technology confusion**, highlights the fact that some software has been designed to make trust decisions based on the assumption that dotless names always refer to trusted hosts on private networks. Technology confusion is demonstrated, historically, by the automatic granting of dotless certificates from Certification Authorities (CAs), the "Intranet Zone" setting in Internet Explorer & Microsoft Windows, and the common use of dotless names to reference internal resources such as file shares.

The study suggests that this inherent trust in dotless names, by users and software, may lead to confusion when handling new Internet facing dotless domains. This confusion can result in unexpected behavior and a misappropriation of trust, ultimately degrading the stability and security of the Internet.

To further address the subjective nature of these concerns, Carve recommends that follow up studies be conducted. One study should be designed and executed to identify specific high-risk names due to the namespace collision introduced by dotless domain names on the Internet. A second study should be performed to specifically quantify the level of human confusion created by the use of dotless names on the Internet.

In the event that applicants are permitted to operate gTLDs in a dotless fashion, Carve recommends that outreach be performed to educate the software development community about the risks associated with trusting dotless names. This document, along with additional case studies and specific software engineering recommendations, can help software developers adapt their applications to a potentially different Internet namespace.

Risks and Mitigations

Carve Systems selected ten risks that it felt were the most representative, or novel, from the risk assessment. Out of these ten issues, two risks could not be backed with adequate supporting evidence, and as such are considered ***not confirmed*** and require further study. The remaining eight risks were confirmed during the application testing phase. For a detailed breakdown of the risks, please see: Appendix A: Risk Listings and Potential Mitigations.

Subjectivity	Description
	Confirmed – risk not mitigated
	Confirmed – risk mitigated
	Confirmed – risk partially mitigated
	Unconfirmed – not confirmable during study

Figure 2: Subjectivity Key

Risk and Mitigation Table			
#	Risk Title	Description	Subjectivity
1	**Internet vs. Intranet resource confusion**	Web Browser classifies dotless names as Intranet sites due to misappropriation of trust.	Confirmed with user prompts in Microsoft Internet Explorer.
2	**"Namespace" confusion due to dotless names already being used internally**	Users cannot determine if dotless names are corporate or 3rd party resources	Not confirmed. Recommend further study.
3	**SSL Client implementation failure resulting in insecure trust decisions**	Anomalous SSL client library behavior due to potentially new use cases that were not considered when dotless domains were isolated to intranet use.	Risk is mitigated based on testing. SSL client libraries that were tested function as expected when parsing dotless domain names. Future, unforeseen, developments and iterations of client libraries and or changes in the technology specification may create additional risks
4	**User confusion in URIs**	Users will be concerned about the legitimacy of dotless names	Not confirmed. Recommend further study.
5	**Public dotless sites gain access to intranet scoped cookies**	Web Browser cookie leakage due to misappropriation of trust	Confirmed
6	**Intranet configured clients, such as browsers and mail clients, may transmit sensitive information over the Internet**	Applications leak data to 3rd party dotless names when corporate resources with same name are disconnected.	Confirmed
7	**SSL certificates for dotless domains already issued by Certificate Authorities**	SSL client misappropriation of trust for previously issued (pre-dotless era) dotless common names	Confirmed due to the existence of intranet certificates that have already been issued and new ones generated during this study. It is important to note that attempts to mitigate this risk are put forth via the updated CA/Browser (CA/B) Forum Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates. The CA/B forum has adopted guidelines on revocation and the sunset of issuing dotless certificates. (https://www.cabforum.org/Baseline_Requirements_V1_1_3_Redline.pdf)
8	**Universal Cross-Site Scripting (UXSS) (https://superevr.com/blog/2012/top-level-universal-xss/)**	XSS on dotless domain completely by-passes Same Origin Policy	Confirmed. Original research posted via https://superevr.com/blog/2012/top-level-universal-xss/
9	**Bonjour/Avahi daemons using new dotless TLD**	Currently, the gTLD ".local" is used by the mDNS system and is incorporated into Apple's Bonjour and Linux's Avahi daemons.	Confirmed. This risk is mitigated by ICANN policy and procedures that classify "local" as a restricted string that cannot be used as a TLD.
10	**Private workgroup and Netbios using dotless domains**	The Windows OS will communicate with dotless Internet resources as if they are local workstation or domain resources.	Confirmed

Figure 3: The Top 10 Risks identified during the Risk Assessment. For full risk information, see Appendix C.

Recommendations

Carve Systems has compiled the following recommendations with a focus on maintaining the security and stability of the Internet in this study:

1) If public dotless names are permitted, disallow potentially dangerous[2] strings from ever being used as dotless TLDs.

   This recommendation mitigates aspects of risk #s 2, 6, 7, 9, and 10.

2) Perform a follow up study that carefully analyzes the namespace collision probability of popular dotless names used on private networks.

   This recommendation mitigates aspects of risk #s 2, 6, 7, 9, and 10

3) Perform a follow up study to quantify the level of human confusion created if dotless names enter the Internet space, after being primarily used by private networks many years.

   This recommendation mitigates aspect of risk # 4.

4) Create awareness among software vendors about the potential for change in the Internet namespace, such that they can prepare their software for a change in trust models.

   This recommendation mitigates aspects of risk #s 1, 3, 5, 6, and 8.

5) Establish guidelines for software and hardware manufacturers to follow when selecting default dotless names for use on private networks. These organizations should use names from a restricted set of dotless domain names that will never be allowed on the public Internet.

   This recommendation mitigates aspects of risk #s 9 and 10.

For more detailed recommendations, please reference Appendix A: Risk Listings and Potential Mitigations

[2] "Dangerous" is used to describe any string that may have unintended, negative, security impact, such as the leaking of sensitive data to Internet hosts. The key property of "dangerous" dotless names (compared to a TLD with a dot) is the namespace collision problem, along with the resultant potential for leaking data. We use the term dangerous to highlight the potential risk, however, the authors of this study note this term is open to interpretation.

Risk Assessment Methodology

Carve took a "People, Process, and Technology" approach to assessing the risk that dotless domains may pose to ICANN, gTLD applicants, and the Internet as a whole. As a starting point, Carve reviewed the SAC053 report, which highlights some of the most apparent risks related to dotless names. After this review, the following steps were taken to execute the risk assessment:

People, Process, Technology, and Dotless Names

Carve specifically considered how people, process, and technology would be affected if dotless domains were allowed on the Internet en masse. This approach was used to consider the specific risk that dotless domains may pose to each category and enumerate risks affecting each.

People: The introduction of dotless gTLDs to the Internet will affect many types of Internet users. For the purpose of this Risk Assessment, Carve will consider Internet users in terms of the following groups:

- Teams within ICANN and those tasked with implementing administrative and technology systems to support dotless domains
- Teams within dotless domain applicants tasked with rolling out web sites and infrastructure utilizing dotless domains
- End-users of systems that utilize dotless domains
- Malicious actors who may try to abuse dotless domains

Carve made educated predictions on how these groups will use and implement dotless domains. For more information on the result of these predictions, and recommended follow up actions, please see the Conclusions and Recommendations section.

Process: Carve considered several different processes in the scope of this assessment; specifically, the gTLD application, approval, and delegation processes.

Technology: Technology supports the people and the process, and in many cases, is indispensable. This risk assessment aimed to identify infrastructure and software components that rely on DNS names to route traffic or make security decisions. This was accomplished via the enumeration of key Internet Architecture components.

Data Routing and Trust

Carve considers the impact of dotless domains in two primary ways: **broken routing models** and **broken trust models**. In the case of routing models, this is meant in a very general sense of data not arriving at its intended destination, and not in the specific definition of routing protocols. For trust models, the meaning is any situation where a trust decision such as, "Is this certificate trusted?", or, "Do I recognize http://ecommerce as a site I trust?", is presented to a technology system or end-user. Carve tried to understand the areas where dotless names may cause security or stability issues into one of these two models.

Methodology Details

The following section explains the details of how Carve built and executed the risk assessment methodology. The first step was the enumeration of Internet architecture. This involved examining the people, process, and technology involved with dotless names, and intersecting that with the most popular, and critical, Internet infrastructure components that use the Domain Name System (DNS), and would thus be impacted by dotless domain names. After enumerating the Internet architecture and defining the application classes for the study, Carve created threat models for each of the application classes. After creating the threat models, application class specific risks were derived and a risk matrix was created.

Enumeration of Internet Architecture

As highlighted throughout this report, people, process, and technology are the three key areas that Carve focused on to understand the risks posed by dotless domain names. By examining widely adopted Internet software, such as web browsers and web servers, mail clients and mail servers, Carve created a list of critical application classes that rely on the DNS system. This definition of application classes was the first step in the implementation of the risk assessment methodology for this study.

Creation of Threat Models for Application Classes

For each defined application class, Carve created a threat model focused on the use of, and reliance on, the DNS system. The threat models were each customized based on the technology use cases, and requirements, along with how the application class used the DNS system. The attack surface was then enumerated. The individual components of each application class were considered (as it related to DNS) and then an assessment of each component of the application class was performed.

Derivation of Application Specific Risks from Threat Models

Following the creation of the threat model, Carve was able to identify specific risks. The risks were then studied and refined, and used to generate test cases to examine how said risks might manifest themselves within real world applications and systems. For more details on the specific hypothesized risks and potential mitigations please see: Appendix A: Risk Listings and Potential Mitigations

Application Testing

Application testing was conducted based on the risks identified during threat modeling. The identified risks were used to draft test cases. The testing was then performed and the results noted. Many test cases were informed by relevant RFCs that documented the de-facto standards of the application classes being tested. It should be noted that the majority of the testing was designed to test "Technology" risks.

Creation and Execution of Application Test Cases

The test cases were designed to test the breadth of the application class in question given the study's time constraints. Certain risks, including some effecting People or Processes, could not be tested within the scope of this engagement. Only tests that could have clear "pass/fail" criteria were designed, and ultimately executed.

Summary of Testing Results

Throughout the risk assessment and testing, there was one broad theme. DNS and server technology has been capable of dealing with dotless names for over twenty years. Dotless names are a core part of many dotless networks. The closer an application class was to end users, such as web browsers, the more potential for risks that would impact the security and stability of the DNS system. This rule was just a generalization, but it tended to fit with most of the risks and testing results.

For detailed testing results, please see: Appendix B: Testing Notes.

Conclusions and Recommendations

After completing the study on the security and stability impact of dotless domain names, Carve has compiled several recommendations. During the study, it became clear that most of the application classes studied currently support dotless domain names. Software that would use dotless names over a private network will also support them over a public network. Based on the three concerns highlighted in the Executive Summary, namespace collision, user confusion, and technology confusion, Carve has the following high-level recommendations.

Namespace Collision Recommendations

In the event that dotless domain names are allowed, Carve suggests that potentially dangerous strings be identified and reserved for use on internal networks only. The criteria for classifying a dotless string as "dangerous" would be how widely the string is used to resolve internal resources on private networks. The more a dotless TLD is used across individual private networks, the greater the potential for negative impact in the event the name becomes publically accessible on the Internet.

One method for generating a list of dangerous strings is to identify DNS requests for dotless names that have leaked to the Internet. Root server data analysis could be used to create a list of leaked dotless names, and this list can be further analyzed based on the frequency that names appear. The leaking frequency should be taken into consideration during the gTLD approval process to make judgments on a string's potential impact on private networks. A high frequency would potentially lead to a string being added to a restricted list or carefully controlled via contractual obligations between ICANN and the applicant.

The DNS Operations Analysis and Research Center posted a blog article (https://www.dns-oarc.net/node/314) that contained data describing "single label" strings that leaked to public DNS servers. A more structured analysis of this data could help to determine what strings should be reserved and/or carry additional risk when used in a dotless fashion. Based on this list, and the "Name Collision in the DNS"[3] study, strings such as localhost, lan, internal, corp, home, belkin, etc are all being leaked to public DNS servers at a relatively high frequency compared to other "single label", or dotless, strings.

User Confusion Recommendations

To address the intranet versus Internet user confusion issue in a less subjective manner, Carve recommends that studies be designed and executed to survey Internet and corporate network users. Carve recognizes that the logistical challenges of conducting such surveys are not trivial. However, it is the best path forward that Carve can recommend for understanding the actual impact of dotless names on users.

While the survey of Internet users would potentially focus solely on the user's interpretation of dotless names, the corporate study could have a less subjective aspect. The corporate study could be conducted with the co-operation of private organizations, and in addition to human surveys, could also take into account the actual use of dotless names on subject private networks. This aspect of the study would take into account an analysis of the subjects DNS zone databases.

[3] http://durban47.icann.org/meetings/durban2013/presentation-dns-name-collision-17jul13-en.pdf

Gauging "human readiness" for dotless names is one of the most difficult areas of this study to quantify. A traditional approach to solving problems with end-users is education. Educating users would entail creating awareness of what a dotless name is and what the "rules" are. It is reasonable to assume that non-technical users have developed informal and intuitive rules about what domain names look like and how they work. Dotless names may constitute a substantial change to the average non-technical user.

Based on Carve's understanding of how information security has evolved over the last decade, it seems inappropriate to simply dismiss the option to foster awareness about dotless names. Users could be informed through a variety of means, such as direct software interaction, and traditional marketing efforts. The exact recommendations would depend heavily on the results of the "user confusion" survey, focusing on Internet users.

Technology Confusion Recommendations

It is not possible for ICANN to individually assess all software components for stability and security concerns related to the potential deployment of dotless names on the Internet. However, ICANN should make available information and recommendations, in the form of open source case studies and white papers, designed to educate the Internet engineering community on the risks associated with creating software that places inherent trust in dotless names.

Tactical Recommendations

The below points were originally expressed in the executive summary. Here, they are expanded upon further.

1) If public dotless names are permitted, disallow dangerous strings from ever being used as a dotless TLD.
Policies and process already exist within ICANN to restrict the strings that can be used as publicly addressable TLDs. The list of restricted strings should be amended to include strings that should not be allowed to function as dotless TLDs as per findings of subsequent studies defined in the above conclusions.

2) Perform a follow up study that carefully analyzes the namespace collision probability of popular dotless names used on private networks.
The goal of this study would be to better understand what dotless names are currently deployed within private networks. The conclusions of this study should detail broad patterns of usage within larger[4] (1000 or more node) networks.

3) Perform a follow up study to quantify the level of human confusion created if dotless names enter the Internet space, after being used primary by private networks many years.

4) Work with software vendors to ensure that software does not make unsafe assumptions about the origin of a host origin based solely on its TLD.
In cases where domain names are used to make security decisions, the underlying logic should be assessed thoroughly. This assessment can be done using standard threat modeling methodologies[5] and software security assessment[6] methodologies. It is difficult to be prescriptive given the wide role domain names play in software and trust, but the general principle outlined here is sufficient. Organizations should consider addressing any identified risks in their software development lifecycle.

The problem of determining how to trust a remote host, based on a name (and a variety of other factors) is a problem that SSL/TLS (along with the X.509 trust model) solves. Other approaches, such as DNSSEC[7] and Moxie Marlinspike's Convergence[8], have been defined as alternate trust models.

5) Home routing equipment should use names from a restricted set of dotless domain names that will never be allowed on the public Internet.
The usage of these restricted names will improve the odds that home routing equipment does not use a name that will become "dotless". This will help eliminate, or at least reduce significantly, potentially sensitive information from leaking, along with addressing other concerns outlined in this study.

---

4 This is a first estimate of what constitutes a "large" network. The actual study may re-define what is a "large" network depending on further setup work.
5 http://msdn.microsoft.com/en-us/library/ff648644.aspx
6 http://www.pearsonhighered.com/educator/product/Art-of-Software-Security-Assessment-The-Identifying-and-Preventing-Software-Vulnerabilities/9780321444424.page
7 http://www.icann.org/en/about/learning/factsheets/dnssec-qaa-09oct08-en.htm
8 http://convergence.io

Appendix A: Risk Listings and Potential Mitigations

Carve enumerated a number of risks as they relate to the application classes under test. These risks were developed by focusing on how specific threats may lead to data routing errors or trust decisions being made incorrectly as related to dotless names. It is important to understand that these risks are a list of potential concerns that may or may not be currently mitigated via systems already deployed. Carve used these risks as a basis to develop test cases to identify if any are manifested within currently deployed technologies or processes. The table below enumerates these risks and highlights what application class they may apply to, and whether they belong within the people, process, or technology bucket.

Risk and Mitigation Table

Risk and Mitigation Table

#	Risk Title	Asset	Area	Description	Recommended Mitigation	Testing Results
1	**Internet vs. Intranet resource confusion**	Browser	Tech	Browsers that treat dotless Internet websites as intranet websites will inappropriately assign more trust to these dotless, public, sites. SAC 053 identified this risk. The problem is only verified to exist in Internet Explorer.	Software must not assume that, because a name is dotless, that it is an Intranet resource. Safe UX and UI interfaces would inform a user when a URI points to an Intranet resource, an Internet resource, or a search query. If this determination cannot be made, software should not make any security decisions on the assumption that a dotless domain is intranet facing. This UI/UX enhancement may be achieved via configured intranet ip ranges within a user's client.	Confirmed via Internet Explorer Intranet profile.

2	**"Namespace" confusion due to dotless names already being used internally**	General	Tech	Dotless domains are already heavily in use by private networks. These dotless domain names are widespread and the scope of the technology impacted is considerable. The confusion of this namespace collision could lead to a variety of attacks and other security failures due to systems assuming they are connecting to trusted resources on an internal network. Attacks range from new phishing scenarios that utilize the inherent expectation that a dotless domain points to an internal resource to rendering commonly configured systems inoperable due to network configuration that may conflict with Internet pointing dotless domains.	One possible, though encompassing, mitigation is for ICANN to not allow dotless domain names. Other approaches include UI and UX improvements to common clients, such as modern web browsers, that make it clear when public (internet addressable) sites are being accessed. This UI/UX enhancement may be achieved via configured intranet ip ranges within a user's client. If this determination cannot be made, UI and UX changes could be implemented to warn users that the dotless domain they are accessing may be internet facing. ICANN can also create a set of domain names that are disallowed from being dotless, such as "mail" or "onion". This study is only preliminary and an exhaustively list of names will require further study, however the general strategy is to come up with a reasonable methodology to evaluate strings as dotless names and what impact that might have to the security and stability of the Internet.	Partially confirmed. Highly suspected based on knowledge of numerous corporate intranets, and raw data from DNS root servers from the namespace collision study. This study documented that a number of queries with non-delegated TLDs are making it to the root servers. These are most likely due to their heavy use on intranet networks. Various technology stack configurations utilize dotless domains to point to intranet hosts.
3	**SSL Client implementation failure resulting in insecure trust decisions**	SSL Client and Server libraries	Tech	SSL clients rely on the common name of certificates, and certain basic constraints in browsers, to make a trust decision. Dotless domain names may short-circuit this and or be issued to untrustworthy entities. Dotless domain names, though commonly used for intranet and internal sites, have not been used on the public Internet. The scale of their deployment may cause SSL clients to make invalid trust decisions.	SSL client libraries must follow any new standards used to mitigate against the issuance of dotless domain name certificates already in the wild. There are currently no standards being developed to address potential dotless certificates already in the wild. However the CA/B forum has issued updated procedure that advises CAs to stop issuing certificates for non-delegated domain names (which includes dotless names).	Verified that current popular CAs did not automatically issue new dotless certificates. However it should be noted that these certificates have been issued in the past, and may exist in the wild. One must consider that with social engineering attacks targeting CAs with lax controls, it may still be possible to obtain a dotless domain certificate. This issue is the subject of another ICANN study and has other mitigation strategies, such as Ballot 96.

4	**User confusion in URIs**	General	People	Users viewing dotless domains in URIs may be confused about the legitimacy of these URIs. Additionally, these URIs may be confused with an intranet URI even though they are publicly resolvable DNS names. Corporate intranets very commonly use these dotless schemes for their internal sites that are served on private IP address space.	UI and UX improvements to common clients, such as browsers, which would make it clear to users when public sites are being accessed. If this determination cannot be made, software should inform the user that the dotless domain being used might be pointed to external/internet hosts. This UI/UX enhancement may be achieved via configured intranet IP ranges within a user's client. In addition, restricted set of domain names that are disallowed from being dotless, such as "mail".	Not confirmed. Requires user survey.
5	**Public dotless sites gain access to intranet scoped cookies**	Browser	People	Internal sites that have names that suddenly become public, may unintentionally receive cookies and other sensitive data from internal corporate, and home network, web applications. More specifically, if a user is accustomed to using an intranet web resource that shares a name of a newly deployed dotless domain, any cookies tied to the intranet dotless domain may be forwarded to the new internet dotless resource. This is due to the fact that a user's browser will scope the cookie to the domain and not be able to determine the difference between an Internet and intranet site. If the user mistakenly accesses the Internet site, their intranet cookies will be forwarded over the public internet. This could lead to rather sensitive intranet and user data to be leaked onto the public Internet.	UI and UX improvements to common clients, such as browsers, which would make it clear to users when public sites are being accessed. If this determination cannot be made, software should inform the user that the dotless domain being used might be pointed to external/internet hosts. This UI/UX enhancement may be achieved via configured intranet IP ranges within a user's client.	Confirmed that cookies will be forwarded based on the dotless domain used. If a user mistakenly accesses an Internet facing dotless domain that collides with a dotless domain for an Intranet resource, cookies set while using the intranet site will be forward to the Internet site.

6	**Intranet configured clients, such as browsers and mail clients, may transmit sensitive information over the Internet**	General	Tech	When users use internal sites, with specific dotless names, they will typically end up with session cookies and persistent cookies. In addition, mail clients, ftp clients, and other pieces of software may be configured to connect to an intranet resource via a dotless domain. These clients will store data specific to the user, such as authentication credentials. Many corporate users now use mobile devices and laptops and take their devices outside of their corporate network. When a user attempts to access these sites or utilize other clients while not connected to the corporate network (via VPN, or direct access), their browsers and other applications will send sensitive data to Internet dotless sites that share the same dotless domain as the Intranet resource. This risk is a product of namespace collision.	UI and UX improvements to common clients, such as browsers, which make it clear to users when public (internet addressable) sites are being accessed. Restricted set of domain names that are disallowed from being dotless, such as “mail” or “onion”. Enforce granular network access on a per-application basis.	Confirmed that clients that utilize dotless domains to access backend resources will not attempt to distinguish between Internet and intranet resources. If a client attempts to connect to a host via a dotless domain, and that domain also has a public IP assigned to it, it may attempt to connect and send information to the public internet host.

7	**SSL certificates for dotless domains already issued by Certificate Authorities**	SSL	Tech	Currently, though it is being sunset by many large trust providers, it is possible to register Intranet certificates that chain back to a trusted certificate authority. With dotless domains, it will not be possible to easily distinguish between a valid certificate, or one that has been previously issued but has not yet expired.	Due to the fact that applying for a gTLD is a relatively involved process as compared to obtaining a normal domain name, and the application process for a gTLD to potentially be used as a dotless domain will incorporate even more checks, it is safe to assume the number of potential dotless domains will be orders of magnitude less than traditional domain names. Therefore trust providers should implement special procedures for issuing dotless domain name certificates. They should also cease to issue dotless Intranet certificates. Carve Systems also recommends, if allowed, for the newly issued dotless domain certificates (the new public ones), that a new basic constraint be added to make it clear the certificate is a valid certificate issued after dotless domain names were allowed by ICANN. This problem is addressed in part by Ballot 96 [9] and the CA/B Baseline Requirements.	Popular CAs have incorporated the CA/B forum update on phasing out dotless domain certificates. There may be lesser known but still trusted CAs that have not fully implemented the CA/B forum recommendations. Therefore the addition of the dotless basic constraints, in addition to verification code in popular applications (browsers, ssl libraries, etc), would help protect users from those rogue certificates that may still make it into the public.

[9] https://cabforum.org/pipermail/public/2013-February/001191.html

8	**Universal Cross-Site Scripting (UXSS) (https://superevr.com/blog/2012/top-level-universal-xss/)**	Browser	Tech	Universal XSS applies to dotless domains and web browsers that elevate privileges of website accessed via a dotless domain. Internet Explorer, for example, removes or lowers a number of security settings when rendering a dotless domain as an intranet property. This lowered security profile violates a number of polices defined in SOP (Same Origin Policy), a set of rules followed by browsers to protect users against potentially malicious web sites. Due to SOP, it would normally be quite difficult for a vulnerability in one website to affect another website. However, when Internet Explorer accesses a dotless website, and enters into the intranet security profile, it would allow certain vulnerabilities on the dotless website to affect ALL other websites the user may access. More specifically, it would allow any XSS vulnerability on a dotless domain to gain access to cookies for other domains, by allowing JavaScript to be run against unassociated domains. This would mean that a bug in a dotless website could be used to target any website a user frequents.	Software must not assume that, because a name is dotless, that it is an Intranet resource. Safe UX and UI interfaces would inform a user when a URI points to an Intranet resource, an Internet resource, or a search query. If this determination cannot be made, software should not make any security decisions on the assumption that a dotless domain is intranet facing. This UI/UX enhancement may be achieved via configured intranet IP ranges within a user's client.	Confirmed threat. Original research posted via https://superevr.com/blog/2012/top-level-universal-xss/

9	**Bonjour/Avahi daemons using new dotless TLD**	Mac/Linux OS	Tech	Currently, the name ".local" is used by the mDNS system and is incorporated into Apple's Bonjour and Linux Avahi daemons. These daemons are used to locate computers on the local network, much the same way users are accustomed to utilizing computer names to access computers on a Windows network. Any mDNS request that makes it to root servers may mistakenly attempt to access hosts that the .local TLD points to.	No TLD or dotless domain name should be "local" or ".local". This name is already reserved.	Confirmed that .local requests make it to the DNS root servers. However the .local TLD is already a reserved TLD, so a public DNS collision should not happen.
10	**Private workgroup and Netbios using dotless domains**	Windows OS	Tech	Netbios, Workgroups, and Windows domains all use dotless "domains" to address other systems in a group. If a computer shares the same name as a proposed dotless domain, users may become confused and connect to public, Internet resources. This may lead to sensitive information, such as Windows credentials, being sent to nodes over the public Internet. In this case, this risk highlights the namespace collision issue within Windows OS networking components.	Default configuration of home networks should use restricted dotless domain sets by default.	Confirmed that windows will attempt to connect to public IPs via the common "whack whack" (\\x.x.x.x\) method.
11	**Malicious dotless domain registration**		Process	Owning and operating a dotless domain requires an organization that is highly responsible and trusted. Malicious operators of dotless domains can use those domains to perform attacks against Internet users.	Appropriate screening measures must be put in place to ensure that potentially malicious operators are never given the opportunity to operate a dotless domain.	Confirmed that the new gTLD Program screening process is indeed rigorous and should address this threat.

12	**Public IP resolution of internal computer names**	SOHO Routers and Networking	Tech	Internal host names for private networks may resolve to public IP addresses, resulting in users submitting confidential information to public Internet sites. This may be possible due to the fact that a number of modern SOHO routers provide their own DNS service to network users, and that these routers also try to catalog hosts on the network. If there are naming conflicts between a host on the network and a potential dotless domain name, the SOHO router may be confused on what destination to forward.	Other approaches include UI and UX improvements to common clients, such as browsers, that make it clear to users when public (internet addressable) sites are being accessed. Create a restricted set of domain names that are disallowed from being dotless, such as "mail".	During testing, it was found that the SOHO router DNS server took precedence to names associated to nodes on the network. Assuming the host uses the DNS server on the SOHO router, and there is a naming conflict, the local host takes precedence.
13	**Proxy / Network Filter Bypass**	Proxies	Tech	Proxies may fail to properly process dotless domain resources, allowing for successful filter evasion attacks. Proxies may treat dotless domain names as trusted resources. Dotless domain names may also cause confusion regarding client proxy configuration. This would be due to the paradigm that the proxies are initially looked up via intranet facing dotless domains. If a user attempts to look up a proxy location via a dotless domain that is registered to an internet address, it may try to connect to a remote host.	Proxy server configurations should properly route traffic from trusted clients to trusted internal resources. This can be accomplished by using appropriate DNS settings. Proxy aware clients should distinguish between trusted proxies on the internal, trusted network, and untrusted proxies that potentially share the same dotless name on the Internet. This can be accomplished by avoiding the use of dotless domains to refer to proxy servers. Proxy configurations should use either fully qualified domain names, or IP addresses.	Proxies behave as expected. DNS resolution will be performed by the DNS client of the proxy operating system. If a client is configured to connect to a proxy identified by a dotless domain that conflicts with the name of an internet facing dotless domain, the DNS server will make the determination on which proxy the client will use. If a host is using a public DNS server, it will most likely cause the client to use the public proxy server.

14	**DNS Protocol implementation vulnerabilities**	DNS Servers	Tech	Technical bugs within a DNS server implementation related to the handling of dotless names may result in memory corruption and other software bugs. This may lead to compromise of DNS clients and servers.	Ensure that DNS server code is developed within a proper SDLC (Software Development Lifecycle) process.	Very low likelihood of happening. Our tests show that all tested DNS servers properly handle dotless domain names.
15	**Same origin policy and cookie binding failure**	Browser	Tech	Developers that improperly bind a cookie, or applications that improperly interpret cookies may make invalid decisions based on unexpected cookie values, causing sites to fail or insecure logic decisions to be made. This has historically been seen during the development of cookies as highlighted via this article. (http://lcamtuf.blogspot.com/2010/10/http-cookies-or-how-not-to-design.html). Early development of cookie usage out right utilized the dots in a domain name to make decisions. Improper assumptions have lead to security failures in the past.	Browser logic must be reviewed to ensure properl handling of dotless domain cookies and requests.	Confirmed that under normal circumstances, all tested browsers understand the concept of a dotless domain.
16	**SMTP servers fail to support dotless email addresses**	SMTP Servers	Tech	SMTP servers attempting to send mail to an address at a dotless domain would likely fail, as SMTP servers may not be configured to support dotless names.	SMTP servers should support dotless names as per RFC5321 section 4.1.2. Mail servers that do not comply with RFC5321 should be upgraded if they need to handle email sent to addresses at dotless domains.	SMTP Servers support the ability to send and receive emails to/from a dotless domain. Dotless names are acceptable as per RFC5321 section 4.1.2

17	**Web page requests resolve to internet hosts instead of proper intranet hosts**	Web Servers	Tech	Web servers that rely on other server resources may end up attempting to access those resources from public Internet sites due to the confusion of the dotless "namespace". For example, a web server attempting to access an intranet database server via a dotless domain name may attempt to access a server on the Internet if there is a dotless domain conflict.	Web server configurations must be reviewed to ensure they do not attempt to access Internet resources.	Informational concern. This concern would require an outright error on the part of an administrator to cause a security incident. This would fall under a misconfiguration leading to a security vulnerability that is independent of dotless domains. This is another instance of the namespace collision issue, as applied to web servers.
18	**SSL verification failure for dotless domains**	Browser	Tech	Sites that rely on SSL may fail SSL verification for dotless domain names. Certificate Authorities are currently phasing out Internal Server Names linked to a public root CA certificate. Browsers in an "Internet" zone may reject SSL certificates for dotless sites. (https://search.thawte.com/support/ssl-digital-certificates/index?page=content&id=AR1809)	Browsers must ensure their certificate validation code is implemented correctly and that dotless domain names are not treated any differently regarding SSL certificate verification.	Informational concern. The hypothesized likelihood of this occurring was quite low. During testing, we found that SSL verification libraries and processes in popular software properly handle dotless domains.

19	**Long form IP addresses similar to dotless domains**	General	Tech	Long form IP addresses (they appear as a large decimal or hexadecimal number), are very similar to dotless domain names. Most applications support long form IP addresses for pointing to hosts. If there are any errors, or insecure logic, in the code handling these IP addresses it could result in any number of undesirable circumstances (misrouted traffic, bad authentication or authorization decisions, etc.) Note: allowing the registration of a dotless domain that is a legitimate, long form, IP address would have unpredictable effects on browsers. This is currently disallowed by ICANN. For example http://2130706433 is the same as http://0x0x7F000001, which is the same as http://127.0.0.1. Registering the long form IP above, as a dotless domain would effectively register the loop back address.	URI processing code must be properly tested and ensure that it properly distinguishes between long form IP addresses and dotless domain names.	Confirmed that browsers do indeed accept long form IP address. In fact, when using a long form IP address in popular web browsers, a DNS request isn't made. The IP is directly used as the destination host. This risk is mitigated given that TLDs are required to be alphabetic only, per RFC 1123. With the exception of IDN TLDs that cannot be numeric only or look like an hexadecimal number either
20	**Email address validation failure for valid dotless domains**	Mail Clients	Tech	Email addresses in the form of user@domain (no dot) may be considered invalid by popular mail clients, thus preventing mail from being sent.	If dotless domain names in email addresses are desired, email clients must support the form of user@domain (no dot) and allow messages to be sent.	During testing, it was found that most mail clients permit the specification of message recipients in the form of user@domain (no dots). However, the Gmail web interface did not. Dotless names are acceptable as per RFC5321 4.1.2

21	**Load balancing traffic routing logic failure on dotless domains**	Proxies	Tech	Load balancers, and many other high availability configurations, will often use dotless domain names (along with a hosts file or other name resolution scheme) to address internal, non-public resources. These load balancers may attempt to use public facing Internet sites.	Load balancer configurations must ensure they do not attempt to access Internet resources. System administrators can accomplish this through proper enforcement of industry standard information security policy, such as network isolation, VLANs, and firewall policy.	Informational concern. This concern would require an outright error on the part of an administrator to cause a security incident. This would fall under a misconfiguration leading to a security vulnerability that is independent of dotless domains.
22	**DNS Query functions improperly resolving dotless domain names**	DNS Client Library	Tech	Though dotless domain names are widespread, they are not as prevalent as domain names with two or more labels. These libraries may have unknown logic flaws in how they deal with dotless domain names, resulting in insecure logic decisions for the consumers of the library.	Libraries must understand that dotless names will be public and ensure they treat dotless domain names no differently from traditional "dotted" names.	Informational concern. The likelihood of this occurring is very low. In fact, during testing, all libraries properly handled dotless domain names.
23	**Spam / Malware detection failure due to dotless domains**	Mail Clients	Tech	Spam and malware detection agents may fail to understand dotless domains in content, and or execute insecurely due to the usage of dotless domain names.	Ensure that spam and malware detection agents understand dotless domain names and do not improperly handle dotless domain names.	Informational concern. The likelihood of this is very low. Malware detection agents use a number of parameters that would still classify and detect malicious content. During our test, the fact that a message originated from a dotless domain did not necessarily flag that message as spam or malware.

24	**Improperly served web pages due to host / virtualhost misconfiguration**	Web Servers	Tech	Invalid, or inappropriate, potentially sensitive, web pages may be served due to virtual host and other web server logic failing with dotless domain names. Due to the heavy use of dotless domains for internal configurations of web servers, it is suspected that a naming conflict may lead to web servers attempting to connect to public resources.	Ensure that proper guidance is provided to administrators that are configuring web servers.	Informational concern. This concern would require an outright error on the part of an administrator to cause a security incident. This would fall under a misconfiguration leading to a security vulnerability that is independent of dotless domains.
25	**Resolver failure with public IP addresses due to local dotless domain names**	DNS Servers	Tech	Publicly available resources may not be resolvable in a number of locations due to local name servers resolving the name to an internal resource. Example: A dotless domain name, such as 'carve' is allowed on the public Internet. A user then connects to the internal Carve network and uses the internal DNS server. The internal DNS server then directs the user to a local resource for the 'carve' name, though the user might have been expecting a different resource. This is another example of the namespace collision problem.	Networks and systems must avoid using restricted names to ensure they are resolvable in the largest number of circumstances.	Informational concern.

Appendix B: Testing Notes

This section details the results of the testing for each application class. The testing methodology for each application class varied, depending upon the technology in question. Each section has a table that details the testing along with its results.

Browsers

The objective of testing popular web browsers was to catalog behavior of browsers that would be important to dotless domains entering the public Internet space. These behaviors were taken into considerations when articulating risks that may be posed to users.

Target Name	Test	Result
Internet Explorer	Default dotless behavior in address bar was tested to catalog how various representations on how a dotless domain may be used in the address of popular browsers, and how the browsers would interpret it.	Dotless Name Alone ('ac'): searched via default search engine Dotless name with trailing dot ('ac.'): Used as dotless name Dotless name with trailing slash ('ac/'): Used as dotless name Dotless name with prepended "http://" ('http://ac'): Used as dotless name
Chrome	Default dotless behavior in address bar was tested to catalog how various representations on how a dotless domain may be used in the address of popular browsers, and how the browsers would interpret it.	Dotless Name Alone ('ac'): searched via default search engine. Unless previously visited, then used as dotless domain. Dotless name with trailing dot ('ac.'): searched via default search engine. Unless previously visited, then used as dotless domain. Note that even when it is used as a search term, Chrome asks the user if they meant to use it as a valid dotless domain. Dotless name with trailing slash ('ac/'): Used as dotless name Dotless name with prepended "http://" ('http://ac'): Used as dotless name
Firefox	Default dotless behavior in address bar was tested to catalog how various representations on how a dotless domain may be used in the address of popular	Dotless Name Alone ('ac'): Used as dotless name. If dotless domain usage fails, used as search term Dotless name with trailing dot ('ac.'): Used as dotless name. If dotless domain usage fails, used as search term

	browsers, and how the browsers would interpret it.	Dotless name with trailing slash ('ac/'): Used as dotless name. If dotless domain usage fails, used as search term Dotless name with prepended "http://" ('http://ac'): Used as dotless name. If dotless domain usage fails, used as search term
Safari	Default dotless behavior in address bar was tested to catalog how various representations on how a dotless domain may be used in the address of popular browsers, and how the browsers would interpret it.	Dotless Name Alone ('ac'): searched via default search engine. Unless previously visited, then used as dotless domain. Dotless name with trailing dot ('ac.'): searched via default search engine. Unless previously visited, then used as dotless domain. Dotless name with trailing slash ('ac/'): Used as dotless name Dotless name with prepended "http://" ('http://ac'): Used as dotless name
Chrome, Firefox, Safari, Internet Explorer	We tested to catalog differences between accessing a dotless domain destined to a local network host vs. an Internet host.	All browsers would access and load a dotless domain hosted from an Internet host and an internal host exactly the same. For Chrome, Firefox, and Safari, the pages would load much like any other FQDN would load. For Internet Explorer, a dotless domain would request to run under the Intranet Zone.
Chrome, Firefox, Safari, Internet Explorer	We tested to catalog differences between the SSL verification of dotless domains vs. FQDNs.	All browsers parsed SSL certificates. If a certificate did not match the exact host name being loaded, SSL errors would be raised. This was independent of a host being visited by a dotless domain name or FQDN.
Internet Explorer	We tested to verify results reported that a public dotless domain name could be used to put Internet Explorer into an Intranet Zone setting.	Visiting a site via a dotless domain name would cause Internet Explorer to request that the site be loaded with the Intranet Zone settings active. This request would be independent of the fact that the dotless domain name points to an Internet host or an intranet host. If this Intranet Setting is activated, potentially untrusted Internet website would run under a lower security setting intended for trusted intranet websites. This leads to the UXSS described here. (https://superevr.com/blog/2012/top-level-universal-xss/)

Chrome, Firefox, Safari	We tested to identify if there are similar "Intranet Zone" settings present within other browsers that are analogous to what is present in Internet Explorer.	We did not find any settings that mimic the Intranet Zone setting of Internet Explorer.
Internet Explorer, Chrome, Firefox, Safari	Review cookie usage. We tested to ensure that cookies would behave the same way with a dotless domain as it would with a FQDN.	All browsers correctly followed the directives of the set-cookie header in an HTTP response to associate cookies to dotless domain names. These cookies were also correctly sent to dotless domain websites.

Web Servers

The objective in testing web servers were to investigate common web server usage patterns and how dotless domain names entering the public Internet space may affect them.

Target Name	Test	Result
Apache, IIS, NGINX	Virtual host configurations. Although the function is named differently, all major web servers allow one server to host a number of web roots based on the hostname used to access the server. We tested to ensure that the use of a dotless domain name vs. a FQDN with more than one label would not cause anomalous behavior.	During our test, we found that the use of a dotless domain name would function in the exact same way a FQDN with more than one label would. The web servers were able to properly parse the dotless domain name and access the correct web root.

Mail Client

The objective in testing mail clients was to investigate if mail clients would be able to handle emails being sent and received form dotless domains.

Target Name	Test	Result
Outlook, Thunder Bird, Outlook Web Access	Sending an email to a dotless domain.	All of these clients would attempt to send an email to a dotless domain. It is interesting to note that although the Gmail web application would not send an email to a dotless domain, Thunderbird, connected to a Gmail account, would attempt to

		send an email to a dotless domain. In addition, the destinations of mail server did not matter. It would function correctly if the mail server existed on the Internet or on a local Intranet host.
Gmail Web Access	Sending an email to a dotless domain.	The Gmail web application did not allow messages to be sent to an email address to a dotless domain.
Outlook, Thunder Bird, Outlook Web Access, Gmail Web Access	Receiving an email from a dotless domain.	All clients would be able to receive an email sent via a dotless domain email address.
Outlook, Thunder Bird, Outlook Web Access, Gmail Web Access	Spam filter behaviors were tested to see if dotless domain names would necessarily flag emails as spam.	Our test shows that the use of a dotless domain did not necessarily flag emails as spam. The spam detection metrics would use many other factors to determine the spam rating for a message.

Mail Servers

The objective in testing mail servers was to investigate if they would forward and receive emails that used a dotless domain.

Target Name	Test	Result
Postfix	Functionality was tested to verify that a mail server would be able to send and receive dotless emails.	We were able to send and receive dotless emails via a default installation of our Postfix mail server.
Exchange	Functionality was tested to verify that a mail server would be able to send and receive dotless emails.	Exchange 2013 behaves in accordance with the most recent SMTP RFC as it relates to the handling of mailbox names. RFC 5321 amends the domain format to support dotless names. See section 4.1.2. Command Argument Syntax

SoHo Routers

The objective in testing SoHo routers were to investigate if any of the advance functionality provided in these devices would necessarily break due to the potential introduction of dotless domains names to the Internet.

Target Name	Test	Result
Linksys N750, Netgear N300	Do any settings in these routers have issues with	No issues were discovered. The features and advanced functionality of these devices vary widely

	dotless domains, this include local DNS servers settings.	between vendors. The two devices we looked at had no issues but the over all consumer market for these devices is very large so our sample size may not show results that match other devices.

Proxies

The objective in testing proxy servers were to catalog the behavior of popular proxies and how they relate to dotless domain use cases. The functionality would be used to properly define risks that relate to proxy software and dotless domains.

Target Name	Test	Result
Squid3 Proxy	Tested squid as both a forward and reverse proxy.	Squid3 behaves properly as it passes off most domain/IP related questions to the operating system (Linux), which behaves sanely, and as expected. Squid3's default configuration file has comments that directly reflect their knowledge of dotless domains. Specifically their use of the *hosts_file* directive and *append_domain* directive. The latter automatically appends a specified domain to requests. We were unable to retrieve cached content for an internal host.
Apache2 w/ Mod_Proxy	Tested apache as both a forward and reverse proxy.	Apache2 behaves just as squid does. It is much closer to the operating system than a web browser so requests for domain/ip resolution are handed off to the operating system, which behaves in an expected and sane way. We were unable to retrieve cached content for an internal host.
IIS	Tested IIS as a reverse proxy.	Set up IIS 7 on Windows Server 2008 as a reverse proxy. No issues encountered as expected. The use of dotless domains names in the intranet zones in Windows is a well-documented use case for many years.
Tor	Review of TOR 'specification' and design information.	Tor does not appear to be a special case with dotless domains. Due to the way traffic is encrypted only the end nodes can see the original application layer requests. The routing information that is decrypted in transit is not applicable to dotless domains.

OS Daemons

The objective in testing OS daemons that are used to resolve dotless domains on internal network were to define how they may behave with the potential addition of dotless domains to the Internet space.

Target Name	Test	Result
MacOS, Linux	Currently, the TLD ".local" is used by the mDNS system and is incorporated into Apple's Bonjour and Linux's Avahi daemons. These daemons are used to locate computers on the local network; much the same way users are accustomed to utilizing computer names to access computers on a Windows network. We tested to verify if '.local' requests are being sent to public root server and could potentially be forwarded internet facing dotless domains.	We confirmed that .local requests do indeed make it to the root DNS servers. However, due to the fact that .local is already a reserved TLD, an inadvertent collision with a .local lookup and a public DNS server should never happen.
Windows	Netbios, Workgroups, and Windows domains all use dotless "domains" to address other systems in a group. If a computer shares the same name as a proposed dotless domain, users may become confused and connect to public, Internet, resources. This may lead to sensitive information, such as Windows credentials, being sent to nodes over the public internet. We tested to see if the popular "mshome" suffix is being sent to public root servers and could potentially be forwarded internet facing dotless domains.	We confirmed that .mshome queries do make it to the Internet root servers. This leads us to believe that Windows internal network traffic relating to Windows computer name resolutions are making their way into the internet space. Any Windows computer name resolution that collides with a potential dotless domain may attempt to connect to a public server. This would occur because Windows services, such a Windows file sharing, works over the Internet just as it would over a local network.

Web Frameworks

Four frameworks were selected for this assessment: Ruby on Rails, ASP.NET MVC4, Drupal, and Java Play. Drupal is a well-known CMS, but also has grown into a mature framework. Due to its wide

deployment, Carve selected Drupal to represent the PHP stack as it is a popular and well-known (and tested) PHP application. The other frameworks represent modern web frameworks on widely deployed technology that new applications would be developed in. This set of frameworks was not meant to be comprehensive, merely representative.

Target Name	Test	Result
RoR	Intranet vs. Internet confusion of internal / LAN resources, such as database servers and caching servers	Ruby on Rails, when configured to connect to a database with a dotless domain name, would attempt to resolve that host using standard DNS libraries, which would result in a failure to locate the database. When the dotless name was added to the database server, the address resolved and the Ruby on Rails application would then successfully connect to the socket listener, listening at the IP address configured in the internal DNS server. This was the expected behavior.
ASP.NET MVC4	Intranet vs. Internet confusion of internal / LAN resources, such as database servers and caching servers	ASP.NET MVC4, when configured to connect to a database with a dotless domain name, would attempt to resolve that host using standard DNS libraries, which would result in a failure to locate the database. When the dotless name was added to the database server, the address resolved and the ASP.NET MVC4 application would then successfully connect to the socket listener, listening at the IP address configured in the internal DNS server. This was the expected behavior.
Drupal	Intranet vs. Internet confusion of internal / LAN resources, such as database servers and caching servers	Drupal, when configured to connect to a database with a dotless domain name, would attempt to resolve that host using standard DNS libraries, which would result in a failure to locate the database. When the dotless name was added to the database server, the address resolved and the Drupal application would then successfully connect to the socket listener, listening at the IP address configured in the internal DNS server. This was the expected behavior.
Java Play	Intranet vs. Internet confusion of internal / LAN resources, such as database servers and caching servers	Java Play, when configured to connect to a database with a dotless domain name, would attempt to resolve that host using standard DNS libraries, which would result in a failure to locate the database. When the dotless name was added to the database server, the address resolved and the Java Play application would then successfully connect to the socket listener, listening at the IP address configured in the internal DNS server. This

		was the expected behavior.
RoR	Cookie binding/setting improperly for dotless domain names	To test this, the Rails application was set up on a dotless name. Variations of the dotless name were tested using both a hosts file and a local name server. The application was configured to set cookies scoped for the specific domain being tested. In all cases the cookies were properly issued as per RFC 6265.
ASP.NET MVC4	Cookie binding/setting improperly for dotless domain names	The ASP.NET MVC4 application was set up on a dotless name. Variations of the dotless name were tested using both a hosts file and a local name server. The application was configured to set cookies scoped for the specific domain being tested. In all cases the cookies were properly issued as per RFC 6265.
Drupal	Cookie binding/setting improperly for dotless domain names	The Drupal application was set up on a dotless name. Variations of the dotless name were tested using both a hosts file and a local name server. The application was configured to set cookies scoped for the specific domain being tested. In all cases the cookies were properly issued as per RFC 6265.
Java Play	Cookie binding/setting improperly for dotless domain names	The play application was set up on a dotless name. Variations of the dotless name were tested using both a hosts file and a local name server. The application was configured to set cookies scoped for the specific domain being tested. In all cases the cookies were properly issued as per RFC 6265.
RoR	Authorization decisions based on domain/URI executing improperly	The rails application was configured with a popular authorization and authentication package. Sections of the application were then fitted with role based authorization code in their controllers. As expected, the application performed properly and executed the authorization code properly on the dotless scoped application.
ASP.NET MVC4	Authorization decisions based on domain/URI executing improperly	The MVC4 application was configured to enforce authorization constraints using the Authorize attribute within the controller Carve setup. As expected, the application operated properly on a dotless name.
Drupal	Authorization decisions based on domain/URI executing improperly	The Drupal application was set up, and configured, on a dotless domain name. The application was then configured to have multiple users with varying roles. All cookies were forced to be scoped to the 'carve' dotless name, which was resolved by an

		internal name server on a private LAN. All authorization decisions were made correctly.
Java Play	Authorization decisions based on domain/URI executing improperly	Natively, the Play Framework does not come with an authentication or authorization framework. Subjectively, one of the most popular authentication plugins was selected and tested. The application properly functioned on a dotless name site.
RoR	URI routing mechanisms that map URIs to code route incorrectly	The rails application was configured to route dotless names differently from the non dotless version of the internal site. The application properly scoped and bound cookies using dotless specific routing instructions.
ASP.NET MVC4	URI routing mechanisms that map URIs to code route incorrectly	The MVC4 application was configured to route dotless names differently from the non dotless version of the internal site. The application properly scoped and bound cookies using dotless specific routing instructions.
Drupal	URI routing mechanisms that map URIs to code route incorrectly	The Drupal application was configured using Domain Access, which allows for the execution of multiple 'sites' from one Drupal application. One dotless, and one regular, domain name were configured. The application properly served all tested URLs for the dotless, and regular, name.
Java Play	URI routing mechanisms that map URIs to code route incorrectly	The play framework supports a url routing mechanism similar to MVC4 and Rails. It guides developers to use a RESTful URL style. However, the play framework did not easily support multiple domain names. Testing was performed separately, with the same application on a dotless name and a regular dotted name. In both instances the application performed the same.

DNS Servers

The objective in testing DNS servers was to understand how they behaved with dotless names. The testing was designed to encompass a variety

Target Name	Test	Result
BIND	DNS Protocol Implementation vulns in handling of dotless domain names	DNS Server was configured with dotless names on multiple record types. Multiple clients, including nslookup, browsers, dig, and other tools were used to query the server. Server was instrumented for

		memory leaks as well as memory corruption errors. No errors occurred.
BIND	Resolver misbehavior due to dotless domains	Multiple resolvers, such as dig, nslookup, web browsers, ssh clients, and Java software were used to test the resolving of dotless names. The dotless name queries were properly returned in all cases.
BIND	Dotless domains for specific record types returns incorrect or invalid data, resulting in security issues for the systems that reply upon them	A battery of record types were set up, and then queried to determine if the record data was properly returned. In all cases the DNS server properly returned the stored data.
DJBDNS	DNS Protocol Implementation vulns in handling of dotless domain names	DNS Server was configured with dotless names on multiple record types. Multiple clients, including nslookup, browsers, dig, and other tools were used to query the server. Server was instrumented for memory leaks as well as memory corruption errors. No errors occurred.
DJBDNS	Resolver misbehavior due to dotless domains	Multiple resolvers, such as dig, nslookup, web browsers, ssh clients, and Java software were used to test the resolving of dotless names. The dotless name queries were properly returned in all cases.
DJBDNS	Dotless domains for specific record types returns incorrect or invalid data, resulting in security issues for the systems that reply upon them	A battery of record types were set up, and then queried to determine if the record data was properly returned. In all cases the DNS server properly returned the stored data.
DNSMASQ	DNS Protocol Implementation vulns in handling of dotless domain names	DNS Server was configured with dotless names on multiple record types. Multiple clients, including nslookup, browsers, dig, and other tools were used to query the server. Server was instrumented for memory leaks as well as memory corruption errors. No errors occurred.
DNSMASQ	Resolver misbehavior due to dotless domains	Multiple resolvers, such as dig, nslookup, web browsers, ssh clients, and Java software were used to test the resolving of dotless names. The dotless name queries were properly returned in all cases.
DNSMASQ	Dotless domains for specific record types returns incorrect or invalid data, resulting in security issues for the systems that reply upon them	A battery of record types were set up, and then queried to determine if the record data was properly returned. In all cases the DNS server properly returned the stored data.

DNS Client Libraries

DNS Client libraries are critical for applications that need to resolve domain names. Client libraries implement the DNS protocol and communicate with DNS resolver to properly query resolvers with whatever DNS queries the client has constructed. Though the technology and protocol is considered very mature, Carve Systems felt no study of dotless names would be complete without examining the current behavior of these systems in the context of this study.

Target Name	Test	Result
Win32 (binary)		
	Query functions improperly resolve dotless names	Simple applications that use the Win32 DnsQuery function was used to perform DNS queries against dotless names. Dotless names were returned as expected.
	SSL and validation records confused with internal names	The goal of this test was to perform SSL certificate validation against dotless hosts on an 'internal' network to ensure that all queries were properly constrained to the internal name and that typical windows clients made secure trust decisions. Carve Systems verified that SSL clients on windows properly deal with certificates created by a dotless trust authority on an intranet.
	Internal names confused with external names	In this case, a simple client was used with an internal name server. The internal, dotless name, properly resolved to the resource on the configured private network. However, when simulating a device migrating to a public network using a different resolver that also responded with a valid DNS result for dotless names, the application would then attempt to connect to whatever IP address was returned by the simulated public IP address. This was the expected behavior, but illustrates one of the namespace collision issues that can occur.
	Memory Corruption bugs due to dotless domains	Throughout the testing, Application Verifier was used to determine if any memory corruption bugs might be present and related to dotless name resolution. No memory corruption issues were identified.
Android		
	java.net.InetAddress fails for dotless domain queries	A simple Android client was implemented that used java.net.InetAddress to resolve dotless name queries. The functionality worked as expected and was able to properly resolve dotless name functionality.

iOS		
	DNSServiceQueryRecord fails or returns invalid records for dotless domains	A simple, simulated, iOS application was created. This application used the DNSServiceQueryRecord function to resolve a dotless name query. The dotless name query resolved as expected.

SSL Client Libraries

The objective in testing SSL client libraries was to ensure that SSL clients properly function, and validate certificates, that are issued for dotless names.

Target Name	Test	Result
Win32 (binary)	Do SSL client libs handle dotless name constraints for various SSL extensions as expected. ie. Domain Cert, Code Signing, CA Certs	A simple test client was used to validate a certificate for an internal certificate authority (trusted by the local machine). The certificate was issued for a private host on the testing network. The host was set up to use a dotless name. The windows ssl client library properly validated the certificate. When the certificate authority that issued the certificate was removed from the trust store, the client properly rejected the certificate as invalid.
Android	Do SSL libs handle dotless name constraints for various SSL extensions as expected. ie. Domain Cert, Code Signing, CA Certs	A simple test client was used to validate a certificate for an internal certificate authority (trusted by the local machine). The certificate was issued for a private host on the testing network. The host was set up to use a dotless name. The windows ssl client library properly validated the certificate. When the certificate authority that issued the certificate was removed from the trust store, the client properly rejected the certificate as invalid.
iOS	Do SSL libs handle dotless name constraints for various SSL extensions as expected. ie. Domain Cert, Code Signing, CA Certs	A simple test client was used to validate a certificate for an internal certificate authority (trusted by the local machine). The certificate was issued for a private host on the testing network. The host was set up to use a dotless name. The windows ssl client library properly validated the certificate. When the certificate authority that issued the certificate was removed from the trust

		store, the client properly rejected the certificate as invalid.
c# (interpreted)	Do SSL libs handle dotless name constraints for various SSL extensions as expected. ie. Domain Cert, Code Signing, CA Certs	A simple test client was used to validate a certificate for an internal certificate authority (trusted by the local machine). The certificate was issued for a private host on the testing network. The host was set up to use a dotless name. The windows ssl client library properly validated the certificate. When the certificate authority that issued the certificate was removed from the trust store, the client properly rejected the certificate as invalid.

Appendix C: Contact Information

Name	Title	Email
Francisco Arias	gTLD Registry Technical Liaison	francisco.arias@icann.org
Russ Weinstein	Panel Coordination Mgr, New gTLD Program	russ.weinstein@icann.org
Mike Zusman	Principal Consultant	mike.zusman@carvesystem.com
Rajendra Umadas	Senior Consultant	raj.umadas@carvesystems.com
Jeremy Allen	Principal Consultant	jeremy.allen@carvesystems.com