International domain names from a multilingualism and security perspective

Fahd A. Batayneh, Jordan

Abstract

From an Internet governance perspective, multilingualism and security have been two of the cornerstone themes since its inception. The security theme addresses topics regarding the Domain Names System (DNS), Public Key Infrastructure (PKI), Internet attacks, security awareness, and policies and legal measures to ensure a safe and secure Internet experience. Security is a very diverse area where multiple topics should be tackled, and ignoring one or more topics while securing other areas would still jeopardise the safety of Internet users.

The DNS goes back to 1973. It was invented as part of the ARPANET project. It is considered to be a critical asset of the Internet. In accessing websites and e-mail addresses, the DNS makes life easier for Internet users since remembering names is much easier than remembering IP addresses. The uniqueness of the DNS comes from the fact that it uses ASCII characters only.

International Domain Names (IDNs) – on the other hand – use Unicode characters, i.e. code points that represent all languages of the world and are not part of the ASCII table. In other words, IDNs are domain names written in native languages rather than in English.

The DNS has its own set of security threats imposed from the underlying layers of the technology itself, and since IDNs are DNS entries, these threats are applied accordingly. While IDNs fulfil the multilingualism and diversity pillars of the Internet governance process, they impose an additional set of security threats – mainly from linguistic characteristics of various languages.

Keywords: Domain Name System; DNS; international domain names; IDNs; security; multilingualism

Introduction

Today's Internet was developed as part of the US Department of Defense (DoD) ARPANET (American Research Project Agency Network) project in the late 1950s. One of the pivotal goals of the project was to create a network that would continue functioning even if other sections of the network were dysfunctional. In other words, the network was designed to reroute network traffic automatically around problems in connecting systems, or in passing along the necessary information to keep the network alive. Thus, from the beginning, the Internet was designed to be robust against denial-of-service (DoS) attacks. Extensive research at major US universities (MIT, Harvard, Stanford, UCB, UCLA …) was conducted to improve this network – mainly for government and military use. No one expected this network to gain the huge momentum and popularity it has gained so far (Wikipedia, 2011).

Over history, three fundamental concepts of Internet security have been concluded: confidentiality, integrity, and availability. Failure of confidentiality results when an outsider reads or copies information; loss of integrity results when the information is modified illegally; while loss of availability results when the information is removed or becomes inaccessible (CERT, 1998).

Authentication (access credentials) and authorisation (privilege to have access credentials) are the main procedures of Internet security system by which organisations make information available to those who want it and who can be trusted with it. When the means of authentication cannot be refuted afterwards, it is called non-repudiation (CERT, 1998).

An Internet security incident is any network-related activity that has negative security implications for the network itself and other net-

This document has been produced by DiploFoundation with the financial assistance of the European Union. The contents of this document are the sole responsibility of the author and can under no circumstances be regarded as reflecting the position of the European Union or DiploFoundation.

works related or connected to it. As mentioned previously, the original goal of ARPANET was to create a network that would function even if some major sections of the network failed or were attacked, thus the robustness against DoS attacks. The ARPANET protocols were designed for openness and flexibility. The usefulness of the network grew as more sites joined ARPANET (CERT, 1998).

The applications on ARPANET were quite simple: e-mail, newsgroups, and remote connection. ARPANET users were a small group who knew and trusted each other. Cliff Stoll at the Lawrence Berkeley National Laboratory in northern California identified the first international security incident in 1986. An international intruder was using the network to connect to the systems in the USA and copy information; this was due to a simple accounting error in the computer records of the ARPANET systems. The first automated security incident – the Morris worm – was reported in 1988 (CERT, 1998).

As the Internet grew and the number of users increased, so did security threats. Vulnerabilities such as malware, spyware, viruses, worms, Trojan horses, DNS cache poisoning, DoS attacks… and many more are part of the today's exciting Internet. Despite the fact that these threats are of huge concern, especially to newcomers and children, some find them to be the excitement behind today's Internet; after all, we do not live in an ideal world!

From an Internet Governance Forum (IGF) perspective, security has been one of the cornerstone themes ever since its inception. The security theme basically addresses topics regarding the Domain Names System (DNS), Public Key Infrastructure (PKI), Internet attacks, security awareness, and policies and legal measures to ensure a safe and secure Internet experience. Security is a very diverse area where multiple topics should be tackled, and ignoring one or more topics while securing other areas would still jeopardise the safety of Internet users.

A study conducted by the UN-ESCWA and League of Arab States (LAS) states that '… the number of Internet users is increasing at a very rapid rate and coming quite close to the two billion mark. The increase in online content and e-services results in an increase in the number of users and vice-versa. However, this puts millions of new and novice Internet users at risk. Accordingly, Internet security has two focus areas: network assets and personal assets. While the former includes hardware, software, and connectivity, the latter include user practices, devices, and data' (UN-ESCWA/LAS, 2010).

The same study also mentions that '… some segments of the Internet society voice out - through discussions on Internet openness - that the Internet should provide a safe haven where individuals can express their own ideas and safely use its services without the fear of harmful impact. In a paradigm for a privacy/safety in the Internet environment, users should be able to post ideas anonymously and have their privacy maintained' (UN-ESCWA/LAS, 2010).

The DNS goes back to 1973. It was invented as part of the ARPANET project. It is considered to be a critical asset of the Internet. In accessing websites and e-mail addresses, the DNS makes life easier for Internet users since remembering names is much easier than remembering Internet Protocol (IP) addresses. In addition, IP addresses could be changed from time to time, while the DNS rarely is. Thus, when using the DNS, it tends to hide any IP address changes. The uniqueness of the DNS comes from the fact that it uses ASCII characters only (Wikipedia, 2011b).

International Domain Names (IDNs) – on the other hand – use Unicode characters, i.e. code points that represent all languages of the world and are not part of the ASCII table (Wikipedia, 2011c). In other words, IDNs are domain names written in native languages rather than in English.

The DNS has its own set of security threats imposed from the underlying layers of the technology itself, thus they are applied to IDNs. While IDNs fulfil the multilingualism and diversity pillars of the Internet governance process, they impose an additional set of security threats – mainly from linguistic characteristics of various languages. The proposed research shall discuss the DNS, IDNs, and the security concerns they both impose. It aims to cover the

large array of security threats out there. It targets everyone involved in the IDN technical, policy, and governance process, such as national governments, the private sector, educational institutes, research firms, and civil society.

The Domain Name System (DNS)

During the early days of the Internet, and after continuous research over the years in an effort to enhance this new technology, new modules were required to make the Internet more robust and easy to access. One of these modules was the Domain Name System (DNS). At the request of Jon Postel, Paul Mockapetris and Kevin Dunlap took the initiative of conducting extensive research on this matter. They started their research in 1982, and in 1988 they came up with their initial model of the system (Mockapetris and Dunlap, 1988). The original specifications were published by the Internet Engineering Task Force (IETF) in Request for Comments (RFC) 882, RFC 883, and subsequent RFCs (Living Internet.com, 2000).

Digging deeper in history, the main intention of the DNS project was to accommodate the expansion of e-mails. During the early days, records were maintained in a centralised hosts.txt file on a server at the Stanford Research Institute (SRI International). This file provided the means of mapping host names to network addresses and vice versa. The format of the file was plain text, human readable. Of course, updates were implemented on the centralised server maintaining the hosts.txt file, and all other servers would pull the updates from the main server. However, one of the main problems of the hosts.txt file is that as networks expanded, so did domain names, thus increasing the size of the maintained file. In addition, when other servers would extract the updates from the centralised server, it would take longer and longer to finish extraction. This led in 1983 to search for new options, and that is when the DNS project saw light (Wikipedia, 2011d).

In 1984, four Berkeley students – Mark Painter, David Riggle, Douglas Terry, and Songnian Zhou – wrote the first UNIX DNS implementation, called *The Berkeley Internet Name Domain (BIND) Server*. BIND is a DNS-resolving software that has gained huge popularity over the years. In 1985, Kevin Dunlap of Digital Equipment Corporation (DEC) significantly rewrote the DNS implementation. Paul Vixie – with assistance from Paul Albitz, Phil Almquist, Fuat Baran, Alan Barrett, Bryan Beecher, Andy Cherenson, Robert Elz, Art Harkin, Anant Kumar, Don Lewis, Tom Limoncelli, Berthold Paffrath, Andrew Partan, Win Treese, and Christophe Wolfhugel – developed BIND 4.9 and 4.9.1. Later on, Paul Vixie left DEC to establish Vixie Enterprises and sponsored the development of BIND 4.9.2, and became the application's principal architect. (ISC, 2011)

What makes the DNS implementation different and more robust than its predecessor – the hosts.txt file – is that domain name entries will be delegated in zone files on distributed machines scattered all around the world. Through the use of special DNS-resolving software (such as BIND), any update that would occur on one machine will be polled to all other DNS servers. In addition, the size of the zone files are extremely small (in KBs or MBs), thus making the time required to transfer the data negligible, especially in our high-speed Internet era.

Not only does DNS support hostname to IP address mapping (or the so-called Forward DNS Resolving), but it also supports IP address to hostname mapping (or the so-called Reverse DNS Resolving). Usually, a DNS hostname is referred to as a Fully Qualified Domain Name (FQDN). Thus, the input to a forward DNS query is an FQDN, and the output is the IP address. As for reverse DNS queries, the input is an IP address and the output is an FQDN.

As technology evolved, so did the Internet. The Internet grew larger as it started to have audience from the public (the so-called Internet community). This led the Internet community to request the formation of a body that could govern the policy aspects of the Internet names and numbers. After extensive discussions and meetings, the Internet Corporation for Assigned Names and Numbers (ICANN) was formed in 1998. ICANN is a not-for-profit organisation based in Marina Del Ray, CA, USA, with offices in Washington DC, Brussels, and Sydney. ICANN cannot control the con-

tent of the Internet, but it has a huge impact on its expansion and evolution (ICANN, 2011a).

One of the main objectives of forming ICANN was to drive the top-level domain (TLD) name industry in the right direction. It focuses on devising policies for the Internet community from the Internet community at large, i.e. decision-makers at ICANN are public Internet users and business entities, and whatever policies come out of it is for the benefit of the Internet and its users (ICANN, 2011a).

TLDs (Root Zone Database, 2011) are classified as:

1. **Generic Top-Level Domains (gTLDs)**
   - They are generic.
   - Examples include .com, .net, .org, .asia, .cat, .info, … etc
   - There exists 21 gTLDs thus far. However, this number could increase in the next few years
   - Registrations are not limited to a country, and they abide to an open policy registration.
   - For any disputes, the Uniform Domain Name Dispute Resolution Process (UDRP) can be used to resolve them (ICANN, 2011b).

2. **Country Code Top-Level Domains (ccTLDs)**
   - They are specific to countries.
   - Two-letter country codes represent each ccTLD. These country codes are extracted from the ISO 3166-1 (ISO, 2011).
   - Examples include .jo, .sa, .ae, .uk, .de, … etc.
   - There exists 240+ ccTLDs.
   - Each registry imposes their own registration policies.
   - For disputes, each registry can devise their own dispute processes, or they can extract part or full of their dispute resolution process from ICANN's UDRP.

While gTLDs are run by entities that build part of their business module around it, ccTLDs vary in their registration policies since they are for countries and not for the whole world. For example, .jo domain names (ccTLD of Jordan) allow domain name registrations for Jordanian entities, as well as alien residents of Jordan. (jo DNS, 2010).

Looking at the domain name industry statistics, .com and .net gTLDs total for 106+ million domain names (VeriSign, 2010). While .com stands at 92.7+ million registrations (makes up for 46% of all TLDs), .net stands at 13.8+ million registrations (makes up for 7% of all TLDs). In addition, the top 10 ccTLDs have domain name registrations of approximately 76.3+ millions with Germany leading the pack at 14+ million registrations (makes up for 7% of all TLDs) and .uk in second place with 9+ million registrations (makes up for 5% of all TLDs) (Shapira, 2011).

One of the hottest topics being discussed within the Internet community is the **New gTLDs** program (ICANN, 2009a). This program aims at introducing new gTLD options; i.e. it aims at expanding the gTLDs on the Internet. The program classifies applications into three categories:

1. **Generic Term TLDs** – These TLDs are related to a generic category such as .shop, .hotel, .music, … etc.
2. **Geographic TLDs** – These TLDs are specific to geographic representations such as cities (.dubai, .tokyo, .ny, .london, .moscow, …), regions (.lac, .latin, .eac, …), or communities (.irish, .zulu, .arab, …).
3. **Brand TLDs** – These TLDs are specific to certain businesses or brands such as .unicef, .canon, .deloitte, … etc.[1]

However, the new gTLDs programme has been the subject of huge debate within the Internet community at large. While many appose to its introduction due to the complexity it might cause to the Internet, others support it since it gives them more options in representing their identities accurately on the Internet, not to mention the business revenue it can make (ICANN, 2009a).

1 For more on possible new gTLD applications, please refer to http://dot-nxt.com/applicants/all

DNS security issues

What make DNS vulnerable to attacks are three main facts: (1) it is an open-for-all protocol meaning that there are no access restrictions; (2) the insecure underlying protocols and lack of authentication and integrity checking of the information within it threaten its proper functionality; and (3) technologies that are used the most are targeted the most. Some of the DNS security threats include (Davidowicz, 1999; Microsoft TechNet, 2005):

- **Foot-printing (information leakage)** – Each domain name has a zone file. A zone file will have entries in the form of sub-domains. The process of foot-printing is to obtain a zone file and extract all sub-domains found in it. These sub-domains might be in the form of entries for PCs, routers, switches, VLANs… etc. With this kind of information in hand, an attacker can figure out the structure of a certain network a domain name belongs to.
- **Denial-of-Service (DoS) attacks** – These attacks take place when an attacker floods a network with recursive DNS queries. When the volume of these queries is so high, the DNS server CPU usage will reach its maximum, thus will be unable to process any further queries. All services that rely on this specific DNS server will become unavailable.
- **Client flooding** – This kind of attack occurs when a client sends out a DNS query but receives a large number of responses from the attacker's name servers. The attack is classified as a success or a failure based on the lack of authentication of the responses, i.e. the response is expected to be from the intended name server, but in reality the response comes out from the attacker's name server. To minimise these kinds of attacks, stronger authentication from the client's side is required.
- **Data modification (IP spoofing)** – The prerequisite for this kind of attacks is a foot-printing attack. Once the attacker has knowledge of the network infrastructure and the range of IP addresses used, he or she can use IP addresses from that network to create data packets and send them out to other users on the network. These data packets can vary in their cause from simply flooding the network with unwanted traffic to data sniffing and serious hacking.
- **Redirection (cache poisoning)** – This takes place when an attacker can redirect DNS traffic from an intended DNS server to a DNS server under the full control of the attacker. One method of doing so is to pollute the DNS cache of the intended server with erroneous DNS data that can force the redirection process. Redirection can happen only when an attacker gains write access to DNS data such as insecure dynamic updates. This would open up the possibility of enormous phishing campaigns and the large-scale theft of passwords, credit-card data, and even access data for online banking. Malicious cache poisoning is commonly referred to as DNS spoofing.
- **DNS dynamic update vulnerabilities** – DNS dynamic updates allow dynamic updating of DNS information contained within a zone file as long as some prerequisites are fulfilled. These updates are usually used to only add and delete DNS records, and are carried out on the primary DNS server. However, the whole concept of dynamic updates is vulnerable to threats such as an IP spoofing the system performing the DNS updates, or compromising the system at large. In either case, an attacker can make an array of attacks ranging from DoS attacks (such as the deletion of records) to malicious redirection (such as changing IP address information for an updated DNS records).

Digging deeper into the DNS security vulnerabilities mentioned above, we can conclude that most of the vulnerabilities – if not all – are IP-based threats. Thus, the major factor behind weaknesses found in the DNS is not solely due to its internal structure, but rather due to the IPs and the protocols it carries. Now the key question is, since the DNS is implemented over IP, and since IP vulnerabilities are not DNS-specific, and since the DNS is a critical asset and a crucial part of the Internet, how do we minimise these threats? The answer to this question is via the deployment and implementation of DNSSec (Davidowicz, 1999).

DNSSec is simply DNS security extensions that aim at raising security levels of DNS with a spe-

cial emphasis on eliminating DNS cache-poisoning. The IETF initiated the work on DNSSec in 1994 as a means to provide security extensions to DNS. The protocol is designed to be interoperable with non-security implementations of DNS - both for clients and servers, is designed for ease of migration and/or update, and is designed to provide backwards compatibility (Davidowicz, 1999).

The fundamental principle of DNSSec is to provide authentication and integrity for DNS. The reason behind that is simply because DNS is a public service, and when data is transmitted in the open, there is a good chance that this data is false. DNSSec uses cryptography in the form of public/private keys to send responses of DNS queries as encrypted data. This ensures that the data received at the receiving end is intact and that it has not been altered or eavesdropped on the way (Davidowicz, 1999).

Despite the fact that DNSSec has been around for a long time, the Internet community concluded its importance when the DNS Cache Poisoning flaw (aka the Kaminsky Bug) surfaced. This bug was first discovered by security expert Dan Kaminsky. In April of 2008, Dan realised that many ISPs had experimented with intercepting return messages of non-existent domain names and replacing them with advertising content, i.e. when a domain name is free for registration and while trying to open that domain name in your web browser, a default web-page would appear. This could allow hackers to set up phishing schemes by attacking the server responsible for the advertisements and linking to non-existent sub-domains of the targeted websites (Wikipedia, 2011e).

DNSSec not only solves the DNS cache poisoning vulnerability, it also solves other kinds of vulnerabilities such as **'man in the middle'** attacks and data modifications (IP spoofing) in authoritative name servers. For a video on DNSSec and the Kaminsky Bug, please refer to Kaminsky Bug.SE (2011).

While the solution to DNS vulnerability was announced on 8 July 2008 (a few days after the ICANN Paris meeting), Dan worked on devising patches with various DNS vendors. In the same month, the USA CERT Team announced that the DNS flaw is a major bug in the DNS protocol itself (Wikipedia, 2011e).

The emergence of international domain names (IDNs)

International domain names – or IDNs for short – are domain names in non-ASCII languages, i.e. rather than writing them using ASCII characters (Wikipedia, 2011b), they are written using non-ASCII characters – or what so called Unicode characters (Wikipedia, 2011c). While English characters are ASCII entries, characters of languages such as Arabic, Chinese, Hindi, Russian, amongst others are Unicode-based languages.

The DNS system understands ASCII only. Thus, in order for DNS to understand Unicode, the Punycode Algorithm (Wikipedia, 2010) was devised to translate Unicode into ASCII and vice versa. The Punycode encoding syntax is defined in IETF's RFC 3492. While Unicode entries are called U-Label (Unicode Label), their ASCII equivalent is called A-Label (ASCII Label) and is usually found in the form of xn--.

IDNs go back to December 1996 when Martin Duerst of the University of Zurich drafted an Internet proposal called UTF-5. At the same time, Prof. Tan Tin Wee of the National University of Singapore carried out similar research. In March 1998, Duerst collaborated with Prof. Wee and his team to enhance their initial work (Peter, 2007).

The first 'General Birds of Feathers' (BoF) meeting on IDNs took place during the INET'98 meeting in Geneva where the Internet community started realizing the importance of this topic (Wikipedia, 2009). Later on, the first 'Technical BoF' meeting on IDNs took place in November 1999 at the IETF Washington meeting. Finally, the first 'Policy BoF' meeting on IDNs took place at the ICANN Melbourne meeting in June 2001 in which the Board IDN Working Group was formed (ICANN, 2001).

From a technical point of view, the IETF has been very active on IDNs. Many working groups have been formed such as the IDN

WG, the EAI WG, and the IDNA WG. Since the majority of the problems related to IDNs are from an application perspective (including security), protocols such as the IDNA2003, IDNA2008, and IDNA2010 were developed to overcome application-level obstacles.

IDNA20XY protocols are IDN applications protocols. These protocols were devised mainly to create language tables for Unicode languages, amongst other technical requirements and needs. Furthermore, these protocols decide what is allowed on which layer of IDN registrations. Each protocol was built upon its predecessor, i.e. IDNA2008 was built on top of IDNA2003, and IDN2010 was built on top of IDNA2008.

From a policy point of view, the Internet community has been heading the initiative under the umbrella of ICANN. IDN guidelines were first created in June 2003, and have been updated to respond to phishing concerns in November 2005. An ICANN working group focused on country code domain names at the top level was formed in November 2007, and promoted jointly by the country code supporting organisation and the Governmental Advisory Committee. Since 2001, many working groups have been created, and many meetings have taken place to arrive at a consensus. In most of its decisions, ICANN would relay back to the IETF – ICANN's right arm in terms of Internet technical implementations. Finally, and during the ICANN Seoul meeting in October 2009, the ICANN board approved the delegation of IDNs for ccTLDs only at the root-servers. ICANN started accepting applications on 16 November 2009 in which applicants would submit their applications through the IDN ccTLD Fast Track Process (ICANN, 2009b). The first IDN strings were approved in January 2010, while the first active IDNs were delegated in the root-servers around April 2010.

When comparing IDN security vs. DNS security, security threats imposed on the DNS apply to IDNs. In addition, IDNs have their own set of unique security concerns imposed from linguistic characteristics of various language sets such as diacritics, variants, and digit mixing.

IDN software issues

As mentioned earlier, domain name resolving software are implemented to resolve ASCII characters, while many software applications are designed to work with ASCII DNS entries only. Thus, there are many software application issues that require special attention. Software applications that require attention include:

1. **Browser applications**
   While most new web browsers support IDNs, many of the old ones – or the so-called legacy web browsers – do not support IDNs directly, i.e. additional plug-ins (if developed) are required.
2. **E-mail client and server applications**
   As we speak, none of the major e-mail server software support IDN e-mails, and the same applies to client software. Afilias – one of the world's leading Internet infrastructure solutions provider – has developed the Global IDN E-Mail software that allows exchange of e-mails using full IDN addresses (Afilias, 2011). However, the software is functional under the Global IDN E-Mail client/server platform; exchanges outside of that platform will not work. Within the IETF, there is the E-Mail Address Internationalization (EAI) working group that is working on technical protocols to widely use IDN e-mails.
3. **Suites of office productivity tools**
   Some of the office suites do not support IDNs. For example, when using IDN hyperlinks within an office document, they do not work properly when clicked on.
4. **Web-based e-mail services, social networking services, blogging and online banking**
   Many – if not all – of the above-mentioned services do not support IDNs.
5. **Look-up tools and command prompts**
   Current look-up tools accept ASCII characters only. If you try to lookup an IDN, no results are returned. In addition, various command prompts (such as **cmd**) accept ASCII characters only; if the language is changed, question marks '?' appear. As of this writing, the only method to lookup IDNs is to enter its A-Label directly in the look-up tool.

6. **DNS registration**
   Most DNS-resolving software – if not all – can resolve ASCII characters only. Thus, when deploying/updating domain name entries in zone files, they must be entered in their A-Label equivalent rather than their U-Label.
7. **Search engines behaviour and optimisation**
   They play an important role in marketing, proper representing, and indexing IDNs. If search engines fail to cope or meet IDN user expectations, this would negatively affect the adoption of IDNs and makes registrants stay away from IDNs under the fear of low ranking their IDN website in the search results.
8. **Software development kits (SDK) and mobile SDKs**
   Mobile applications have become very popular. There are thousands of applications out of which many use domain names in the background to fetch/manipulate data. An obvious example is RSS feed application.
9. **Web hosting solutions providers**
   Examples include hosting automation applications (cPanel, Plesk). End-users expectations for provisioning IDN hosting packages should be as simple as what it does with ASCII domain names.
10. **SSL/digital certificate providers**
    An IDN should be easily signed and be good enough for companies to use in e-commerce.

While many of the issues addressed above have not been addressed in IDNA2003 protocol, IDNA2008 and IDNA2010 protocols have been developed to resolve many – if not all – of them.

IDN linguistic issues and security threats

From a technical point of view, security threats imposed on the DNS apply to IDNs in the same manner since they both are domain names. In addition, and due to the linguistic characteristics that vary amongst different languages, some languages impose extra security concerns.

The visual display of IDNs expose end-users to significant threats, including an increased exposure to phishing attacks and visual spoofing. These attacks can have the side effect of compromising brand reputation and consumer trust. IDN-based visual spoofing attacks have been demonstrated through research and also seen in the wild, yet the adoption of defensive methods has been widely varied across user-agents and other applications. Registries and registrars are in a unique position to counter this threat given the proper tooling. However, proper tooling requires proper understanding of the various linguistic issues of all languages and scripts.

Before talking about IDN linguistic security issues, it is worth mentioning the difference between a script and a language. While a script can include several languages, a language is unique in its nature. For example, the Arabic script consists of languages such as Arabic, Farsi, Urdu, and Jawi. Common findings between these languages include a set of characters with the same Unicode points. However, numerals vary amongst these different languages as each language has its own numeral set.

Since each script and/or language has their own set of security issues to handle, they all revolve around several generic themes. Some of these themes include: (ESCWA, 2008; 2009; Microsoft MSDN Library, 2011)

1. **Character variants**
   Variants are characters that look alike and/or are pronounced alike. However, these variants differ in their Unicode point representation. For example, in the Arabic script we have the Arabic "ك" character (Unicode 0643) and the Urdu "ک" character (Unicode 06AA) (The Unicode Consortium, 2011). Both characters are pronounced alike, and they are visually identical when they are set at the beginning or in the middle of a word. However, when using different keyboards, each character is considered to be different, and this is where the problem occurs. When registering IDNs, one should register all possibilities to accommodate all variants within a script, and there are several ways

to do so. While some register all possibilities, others are working on normalization techniques; i.e. algorithms that consider all variant characters within a script to be – virtually – one code point.
Some IDN software solution providers integrate an online keyboard with their registration systems so that when an IDN registrant wants to register an IDN, they can use the IDN online keyboard to ensure proper registrations. Others – such as Google (2011) – have their own online keyboards so that one can access their native language from anywhere in the world especially when one has no access to native keyboards.

2. **Digit mixing**
   As we all know, there are several numbering systems. However, the most diversely used system – commonly known today as the Western Numbering System (WNS) – is the 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. While it has been agreed upon by the Internet community that digit mixing is prohibited in IDN registrations – at least for now – it has also been agreed upon that the WNS should be used as a unified system by all languages. However, some languages require that their own numbering system be used within IDN registrations since they are part of their language characteristics; thus the need for digit mixing. For example, the word Rama 'امار' in Jawi means butterfly. The plural of Rama is 'Rama Rama' امار امار ('butterflies' in English). However, the plural is written as '٢امار'. '٢' is the number 2 in the Arabic numbering system. If one wants to register this domain name using the agreed-upon conventions, it would be registered as امار2, and this would violate the Jawi linguistic characteristics, thus loosing the core value behind IDNs; i.e. Internet multilingualism and diversity.
3. **Diacritics**
   These are linguistic cosmetics that make the language richer. In Arabic, there are several diacritics that make the Arabic script richer and more pleasant visually when used. While diacritics are very important to many languages, it is agreed upon within the Internet technical community, headed by the IETF, that they should not be used in IDNs – at least for now. However, and due to their high importance, the IETF is working on devising solutions to overcome this matter. It is worth mentioning here that diacritics are considered characters with their own Unicode points. Thus, when they are used, the A-Label representation would differ when using vs. not using diacritics.

As stated previously, these linguistic themes are common to all languages and scripts. However, there are some specific requirements for each language. For example, Cyrillic contains 11 characters that are identical or nearly identical to Latin counterparts; i.e. Cyrillic letters а, с, е, о, р, х, у, В, Н, К, М, and Т have counterparts in the basic Latin alphabet and look close or identical to a, c, e, o, p, x, y, B, H, K, M, and T. In addition, Cyrillic 3, Ч and б resemble the numerals 3, 4 and 6 (Wikipedia, 2011f).

Another example is the Greek language in which some characters are similar to counterparts in other languages. For example, Greek letters κ and ο look similar to Cyrillic к and о, Greek τ can be similar to Cyrillic т in some fonts, and Greek β and ς can be a substitute for German ß and ç in some fonts (US-CERT, 2008).

A further example would be from the Armenian language where some Armenian alphabet can contribute critical characters such as ց, հ, ո, օ, զ, ս which look like the Latin g, h, n, o, q, and u, յ which resembles j, and բ which can either resemble p or f depending on the font. Also, two letters in Armenian (ՁՀ) also can resemble the number 2, while another (Վ) sometimes resembles the number 4 (Wikipedia, 2011f).

In Hebrew, only three letters can reliably be used: samekh (ס) which sometimes resembles o, vav with diacritic (וִ) which resembles an i, and heth (ח) which resembles the letter n. It is worth mentioning here that spoofing is at minimum when using Hebrew IDNs not just because there are 3 similar letters, but also the fact that Hebrew is a right-to-left language (Wikipedia, 2011f).

Finally, the Arabic script has a few characters that are identical to the Latin script. Some of the letters include the Alf ا which looks like the number 1 or letter l, the Arabic one ١ also looks like the number 1 or the letter l, the Arabic

five ٥ or the letter ٥ looks like the number 0, the Arabic nine ٩ looks like the Latin 9, the Arabic eight ٨ looks like the Greek Λ. However, the threat of the Arabic language is minimum as well since it is a right-to-left language.

In conclusion, we can see that not only IDNs impose technical security threats found in the core structure of the DNS, but they also impose many linguistic threats since different languages vary in their linguistic characteristics. These linguistic threats lead to phishing attacks and visual spoofing.

Technical findings of combating IDN security threats

While no one is safe on the Internet, there are specific practices that could be followed to understand this new technology (and any other new technology for that matter) and overcome the threats it imposes. Some of these practices include: (US-CERT, 2008)

1. The uniqueness of IDNs vs. ASCII DNS is the xn-- tag. While ASCII domain names do not start with this tag, all IDNs do have this tag when represented in their A-Label format.
2. Type a URL rather than following a link. This practice will ensure you that you are visiting the intended webpage rather than a malicious redirected one.
3. Since IDNs are one of the hottest Internet topics today, many new software and browser plug-ins are being introduced. And since IDNs and web browsers are like sugar and cake, ensuring an up-to-date browser and plug-ins ensures the highest levels of security.
4. Installing plug-ins that help Internet surfers differentiate between IDNs and ASCII domain names. For example, one can install the **IDND** (Mozilla Corporation, 2010a) plug-in on their Mozilla Firefox web browser. This plug-in will show you the nature of the URL of the website being visited (TDN (Traditional Domain Name), IDN (International Domain Name), or IP Address). Furthermore, there is another plug-in called **WOT** (Web of Trust; Mozilla Corporation, 2010b) that connects to a centralised database consisting of ratings of websites on the Internet. Users can also rate a website.

While the guidelines mentioned above will not prevent IDN security threats on the Internet, understanding them and implementing them wisely can reduce these threats significantly.

Current findings of IDN ccTLDs and the fast track process

Looking at the IDN ccTLD Fast Track Process webpage (ICANN, 2009b), we can conclude the following (as of 20 February 2011):

1. ICANN has received in total 33 requests representing 22 languages.
2. Twenty-three strings representing 23 countries have passed the string evaluation phase.
3. Sixteen different languages from 16 different scripts have passed the string evaluation phase.
4. The IDN ccTLDs from Saudi Arabia, UAE, Egypt, and Russia were the first to be delegated in the DNS root servers.
5. Sri Lanka was the first country to apply for two IDN strings in two different languages.
6. Arabic based IDN requests represent the majority as 11 Arabic IDN ccTLDs have been string-approved, 8 of which have been delegated in the DNS root servers.
7. The application from India represented seven different languages with the Indian administration to request further languages later on.
8. One of the issues facing the Chinese language – traditional and simplified Chinese – was the orthographical variation. While these languages required further string evaluation, they passed it. Countries such as China, Hong Kong, and Taiwan had to go through the orthographical variation technique (CNNIC, 2010).

Russia opened the registration of its IDN ccTLD in Cyrillic under .рф to the public on 11 November 2010 (Coordination Center for Russian TLDs, 2011). During the first hour, 36 607 new domain names were registered, during the second hour 43 054 new domain names were registered, and during the third hour 41 456 new domain names, easily surpass-

ing the benchmark of 120 000+ new IDNs. By the end of day one 240 000+ new IDNs under .pф were registered. As of this writing, domain names under .pф stand at 755 716 registrations.

China started registering IDNs under Simplified and Traditional Chinese under .中国 and .中國 way before ICANN opened the registration under its fast track process (CNNIC, 2010). However, Chinese domain names were resolvable within China only. At the time both Chinese IDN TLD were delegated in the DNS root servers, more than two million domain names were already registered.

Jordan's IDN ccTLD under .الاردن (.alordun) was string-approved on 22 April 2010, and was delegated in the DNS root servers on 22 August 2010. There was a sunrise period for governmental entities for two weeks (17–28 October 2010) and a landrush period for trademarks for four weeks (7 November –3 December 2010), followed by an open registration on 19 December 2010. As of this writing, Jordan has 92 IDNs registered.

Comparing IDNs in Russia, China, and Jordan, we can conclude that there are mixed feelings for them. While the Chinese have been using it for quite some time now (showing its huge importance to the Chinese community), the Russians felt its importance when their IDN ccTLD was delegated in the DNS root server, and the Jordanians have started to feel its importance, but at a slow momentum.

What more is to be done?

While IDNs are not applicable to all languages and communities, those communities that are in need of it find it an extremely valuable resource to drive the Internet forward within their communities. Cases related to China and Russia show how important those communities find IDNs. However, in order to further convince those communities that IDNs are a driving force for their local Internet, they must be aware of the security issues that surround IDNs. Thus, security awareness is a key factor.

While the IDNA protocols have resolved browser discrepancies, there are many issues that need to be resolved. Issues such as global IDN e-mail usage (whether clients or servers), SSL certificates, search engines and optimization, software development kits, lookup tools, Unicode-aware registration software, web-hosting solutions, along with linguistic issues need to be resolved. Once solutions are devised for these pending issues, further enhancing them and coming up with newer protocols remain a driving need.

Looking at the current IDN software market, we can conclude a lack of diversity in IDN software available. In addition, much software is community-specific, especially when it is linguistic related. Furthermore, keyboard diversity among the same script remains a key obstacle, i.e. using different language keyboards within the same script. Thus, we can conclude that there is a long way before IDNs are widely usable.

Summary

The DNS goes back to the early 1970s when the researchers at ARPANET found a need to deal with names rather than dealing with IP addresses. While the DNS has been functional using the ASCII character set ever since it was anticipated until recent, the Internet community found IDNs to be a driving force in assuring diversity on the Internet, not to mention that it is a driving force in increasing Internet penetration and outreaching as many communities as possible.

The year 2010 was a milestone year for the Internet as the first IDNs were delegated in the DNS root servers with Arabic and Chinese IDNs leading the way. While various reactions were made by various communities, the whole concept of introducing Unicode characters on the Internet names brought joy to many communities; especially those who are not at ease using languages other than their native language.

Despite IDNs being an exciting new technology on the Internet, they have a long way to go – especially from a linguistic and security perspective. Many applications do not fully support them as there are many linguistic and security issues that need to be resolved.

References

1. Afilias (2011) *Global IDN E-Mail.* Available at http://ادارة.يلودلا-ديربلا/horde/imp/login.php [accessed 13 April 2011].
2. CERT (1998), *Security of the Internet.* CERT Coordination Center. Available at http://www.cert.org/encyc_article/tocencyc.html [accessed 13 April 2011].
3. CNNIC (2010) *Implementation Plan of .中国 (xn--fiqs8S) and .中國 (xn--fiqz9S)* Available at http://www.cnnic.cn/html/Dir/2010/06/12/5852.htm [accessed 13 April 2011].
4. Coordination Center for Russian TLDs (2011) *Domain Names .рф Statistics.* Available at http://кц.рф/en/statistics/rfdomains.php [accessed 13 April 2011].
5. Davidowicz D (1999) *Domain Name System (DNS) Security.* Available at http://compsec101.antibozo.net/papers/dnssec/dnssec.html [accessed 13 April 2011].
6. ESCWA (2008), *Report of the 3rd Expert Group Meeting on Global Harmonization of Arabic Script use in Domain Names,* Economic and Social Commission for Western Asia Cairo, 8/9 November 2008.
7. ESCWA (2009) *Report of the 4th Expert Group Meeting on Global Harmonization of Arabic Script use in Domain Names,* Economic and Social Commission for Western Asia Amman, 1-3 April 2009.
8. Google (2011) *Google Online Keyboard.* Available at http://www.google.com/webhp?hl=ar [accessed 13 April 2011].
9. ICANN (2001) *Internationalized Domain Names (IDN) Committee.* Available at http://www.icann.org/en/committees/idn/ [accessed 13 April, 2011].
10. ICANN (2009a) *New gTLDs Program.* Available at http://www.icann.org/en/topics/new-gtld-program.htm [accessed 13 April 2011].
11. ICANN (2009b) *IDN ccTLD Fast Track Process.* Available at http://www.icann.org/en/topics/idn/fast-track/ [accessed 13 April 2011].
12. ICANN (2011a) Internet Corporation for Assigned Names and Numbers. Available at http://www.icann.org/ [accessed 13 April 2011].
13. ICANN (2011b), *Uniform Resolution Dispute Process (UDRP).* Available at http://www.icann.org/en/dispute-resolution/ [accessed 13 April 2011].
14. Internet Systems Consortium (ISC), *History of BIND Software Development.* Available at http://www.isc.org/software/bind/history [accessed 13 April 2011]
15. ISO (2011) International Organization of Standardization *ISO 3166-1 Decoding Table.* Available at http://www.iso.org/iso/iso-3166-1_decoding_table [accessed 13 April 2011].
16. jo DNS (2010) *.jo Registration Policy.* Available at http://www.dns.jo/Registration_policy.aspx [accessed 13 April 2011].
17. Kaminsky Bug.SE (2011) *Kaminsky Bug Video.* Available at http://www.kaminskybug.se/movie_en/ [accessed 13 April 2011].
18. Living Internet.com (2000) *Domain Name System (DNS) History.* Available at http://www.livinginternet.com/i/iw_dns_history.htm [accessed 13 April 2011].
19. Microsoft MSDN Library (2011) *Security Considerations: International Features.* Available at http://msdn.microsoft.com/en-us/library/dd374047%28v=vs.85%29.aspx#SC_intl_dom_names [accessed 13 April 2011].
20. Microsoft TechNet (2005) *New Security Information for DNS.* Available at http://technet.microsoft.com/en-us/library/cc783606%28WS.10%29.aspx [accessed 13 April 2011].
21. Mockapetris P and Dunlap K (1988) Development of the Domain Name System. *Computer Communication Review 18(4)*, pp. 123-133.
22. Mozilla Corporation (2011a) *IDND 1.6.0.* Available at https://addons.mozilla.org/en-US/firefox/addon/idnd/ [accessed 13 April 2011].
23. Mozilla Corporation (2011b) *Web of Trust - Safe Browsing Tool.* Available at https://addons.mozilla.org/en-US/firefox/addon/wot-safe-browsing-tool/ [accessed 13 April 2011].
24. Peter I (2007) *History of Internationalized Domain Names.* Available at http://ianpeter.wordpress.com/2007/05/10/history-of-internationalised-domain-names/ [accessed 13 April 2011].
25. Root Zone Database (2011) *Top-Level Domains (TLDs).* Available at http://www.iana.org/domains/root/db/ [accessed 13 April 2011].
26. Shapira I (2011) Rush is on for Custom Domain Name Suffixes. *The Washington Post*, 2 February 2011. Available at http://www.washingtonpost.com/wp-dyn/content/article/2011/02/06/AR2011020603940.html?wpisrc=nl_tech [accessed 13 April 2011].
27. The Unicode Consortium (2011) *Unicode 6.0 Character Code Charts.* Available at http://www.unicode.org/charts/ [accessed 13 April 2011].
28. UN-ESCWA/LAS (2010) *Regional Roadmap for Internet Governance in Arab Countries.* Draft Document ver. 2.0, pp. 18. Available at http://isper.escwa.org.lb/isper/Portals/0/Repository/Roadmap_IG_2.0.pdf [accessed 13 April 2011].
29. US-CERT (2008) *Understanding Internationalized Domain Names.* Available at http://www.us-cert.gov/cas/tips/ST05-016.html [accessed 13 April 2011].
30. VeriSign (2010) *The 2nd Quarter 2010 Domain Name Industry Brief.* Vol. 7, Issue 3. Available at http://www.verisign.com/domain-name-services/domain-information-center/domain-name-resources/domain-name-report-sept10.pdf [accessed 13 April 2011].
31. Wikipedia (2009) *Birds of a Feather (Computing).* Available at http://en.wikipedia.org/wiki/Birds_of_a_Feather_%28computing%29 [accessed 13 April 2011].

32. Wikipedia (2010) *Punycode*. Available at http://en.wikipedia.org/wiki/Punycode [accessed 13 April 2011].
33. Wikipedia (2011a) *History of the Internet*. Available at http://en.wikipedia.org/wiki/History_of_the_Internet [accessed 13 April 2011].
34. Wikipedia (2011b) *ASCII*. Available at http://en.wikipedia.org/wiki/ASCII [accessed 13 April 2011].
35. Wikipedia (2011c) *Unicode*. Available at http://en.wikipedia.org/wiki/Unicode [accessed 13 April 2011].
36. Wikipedia (2011d) *Domain Name System*. Available at http://en.wikipedia.org/wiki/Domain_Name_System [accessed 13 April 2011].
37. Wikipedia (2011e) *Dan Kaminsky*. Available at http://en.wikipedia.org/wiki/Dan_Kaminsky [accessed 13 April 2011].
38. Wikipedia (2011f) *IDN Homograph Attack*. Available at http://en.wikipedia.org/wiki/IDN_homograph_attack [accessed 13 April 2011].

Annex A: List of acronyms

A-Label – ASCII Label
ARPANET – American Research Project Agency Network
ASCII – American Standard Code for Information Interchange
BIND – Berkeley Internet Name Domain
BoF – Birds of Feather
ccTLD – Country Code Top-Level Domains
CERT – Computer Emergency Response Team
DoD – Department of Defense
DoS – Denial of Service
DNS – Domain Name System
DNSSec – DNS Security Extensions
EAI – E-Mail Address Internationalization
FQDN – Fully Qualified Domain Name
gTLD – Generic Top-Level Domain
ICANN – Internet Corporation for Assigned Names and Numbers
IDN – International Domain Names
IDNA – International Domain Names in Applications
IETF – Internet Engineering Task Force
IGF – Internet Governance Forum
IP – Internet Protocol
PKI – Public Key Infrastructure
RFC – Request for Comments
SDK – Software Development Kit
SSL – Security Socket Layer
TDN – Traditional Domain Name
TLD – Top Level Domains
U-Label – Unicode Label
UDRP – Uniform Dispute Resolution Process
WG – Working Group
WOT – Web of Trust