

MTR090302

MITRE TECHNICAL REPORT

Securing multicast DNS

Study into the feasibility of trusted multicast DNS for service discovery

November 2009

Michael John Kristan

Sponsor: MITRE
Dept. No.: E146
Classification: Unclassified

Contract No.: N/A
Project No.: 20AAV140-F1
Caveats: None

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

Approved for public release; distribution unlimited.

©2009 The MITRE Corporation. All Rights Reserved.

MITRE
202 Burlington Road
Bedford, Massachusetts 01730

Version History

<i>Version</i>	<i>Name</i>	<i>Date</i>	<i>Comments</i>
1.0	Michael Kristan	November 10, 2009	Defined terms, removed watermark, and submitted for public release
0.6	Wendy Swirnoff	October 5, 2009	Added acronym section
0.5	Michael Kristan	October 3, 2009	Incorporated feedback from Chris Niessen and Erin Connors
0.4	Wendy Swirnoff	October 2, 2009	Reviewed and edited by MITRE Corporate Communications
0.3	Michael Kristan	September 28, 2009	Incorporated feedback from Erin Connors
0.2	Michael Kristan	September 27, 2009	Added to sections 3.2, 4.1. Incorporated feedback from Jonathan Wielicki
0.1	Michael Kristan	September 21, 2009	Initial draft

Abstract

This paper looks at how developers can use open and freely available technology to provide a level of identity security and trust to service discovery in an untrusted ad-hoc network environment. Initial research shows that using multicast Domain Name Services (DNS) when coupled with X.509 machine-issued certificates gives an adequate level of identification when new computers wish to find and use services across a network. Further research is necessary to generalize the problem and to explore caveats.

Table of Contents

1	INTRODUCTION	1
1.1	SCENARIO	1
1.2	TARGET AUDIENCE	2
2	BACKGROUND.....	3
2.1	DHCP VS. LINK-LOCAL ADDRESSING.....	4
2.2	MULTICAST DNS	4
2.3	SECURE DNS (DNSSEC)	5
2.3.1	<i>DNSSEC lookaside validation</i>	5
2.3.2	<i>Security concerns in DNSSEC</i>	6
2.4	PKI CERTIFICATE DNS RECORDS.....	6
2.5	BONJOUR (DNS-SD) AND ZEROCONF.....	6
2.5.1	<i>How query/response works in DNS-SD</i>	7
2.6	UNIVERSAL PLUG AND PLAY	7
2.7	DISTRIBUTED REGISTRIES	8
2.8	INTERNET PROTOCOL VERSION 6 (IPV6).....	8
2.9	SECURE IP (IPSEC)	8
3	FINDINGS.....	9
3.1	SCOPE OF PROBLEM	9
3.2	WHY DNSSEC WILL NOT WORK.....	9
3.3	CERT RECORDS AS A SOLUTION	9
3.3.1	<i>Trusted root certification authorities</i>	10
3.3.2	<i>User certificates vs. machine certificates</i>	11
3.4	SUPPORT IN CURRENT TECHNOLOGY	11
4	CONCLUSION	13
4.1	NEXT STEPS	13
	REFERENCES AND BIBLIOGRAPHY.....	14
	ACKNOWLEDGEMENTS	18
	ABOUT THE AUTHOR	18
	MITRE INNOVATION PROGRAM	18
	LIST OF ACRONYMS	19
	DISTRIBUTION LIST.....	21

List of Figures

Figure 1 – The OSI model (3).....	3
Figure 2 – Multicast DNS traffic while running iTunes™ on a public network.....	5
Figure 3 – Example of a DNS-SD query in Microsoft Windows XP (16).....	7
Figure 4 – Client/server interaction in service discovery with certificates.....	10
Figure 5 – Windows XP does not support CERT records in nslookup	12

1 Introduction

With the advent of pervasive and ubiquitous computing, society has come to depend on technology as part of daily life. Advances in human factors engineering and pioneering work by high tech companies have made computers extremely easy to use. What the average user does not realize, however, is the amount of complexity that goes into even the simplest of tasks. As a result of design patterns, development techniques, and advances in research, some of those complexities are abstracted out in the form of application programming interfaces (APIs) or lower level processes that leave the developer or the user to focus on less mundane tasks.

In the field of network services, developers are starting to reach the stage of making these services packaged and user friendly. Real-world examples include printers that automatically appear in a shared network, or music libraries that become instantly discoverable and sharable. (1) In both of these examples, no knowledge of configuration parameters is needed to take advantage of them, making it easier for the end user to interact with these components and allowing the user to complete the task with minimal effort and without knowledge of underlying details.

Malicious users are frequently overlooked in these environments. Sadly, cyberspace has grown into an untrusted environment where rogue entities passively and actively try to exploit holes in a network for their own personal gain. Because networks were originally built to share information and not to thwart adversaries' attempts to break the infrastructure, problems occur. (2) Therefore, as technology advances it is imperative that security is not overlooked when planning new protocols, services, or systems.

This paper explores ways to secure service discovery, especially in situations where a fluid, ad hoc network topology exists. The security needs to do a good job at making sure the systems a local computer is interacting with are trusted. It must ensure this in such a way that it is not overly obstructive and does not prevent the end user from doing his or her task. This balance is hard to achieve. In the end, however, there is a desire to prevent situations, such as malicious traps existing on the network, in which an unsuspecting user tries to take advantage of a service on a rogue computer that results in the user's computer being attacked or infected by a virus or root kit.

1.1 Scenario

To scope the problem of trusted service discovery, we can build a simple scenario. . Two computers join a local area network. Each computer is operated by a user who chooses to interact only electronically (i.e., the users do not talk to each other). One computer has a service available, such as instant messaging or file sharing. The other computer is looking to take advantage of this

service. The two computers are both owned by the same company and have the same configuration management and security policies. The computers are connected to an untrusted network, such as a wireless hot spot at a coffee shop. Before allowing the computers to interact, both users want to ensure the other computer is who it says it is.

A non-technical example of this scenario would be a person visiting a popular night club. After looking at a yellow pages entry for local clubs, a stranger approaches a particular club, wishing to dance and socialize. The night club has an age requirement, so the guard at the door needs to verify the stranger's age. The stranger produces a photographic ID issued by the local registry of motor vehicles that verifies the stranger's age. In turn, the stranger also verifies the club's validity based on posted occupancy permits, liquor licenses, and fire inspection certificates. The stranger then enters the club and the transaction is completed.

1.2 Target audience

This paper is targeted to a developer familiar with concepts in service discovery. The reader does not need to be an expert in Domain Name Service (DNS), multicast, or Public Key Infrastructure (PKI) certificates. The background section provides a quick overview of the technologies researched in this paper. The bibliography lists request for comments (RFCs) and journal articles for further reading.

2 Background

To assess possible solutions for the trusted service discovery problem, we examined a large selection of technologies to determine their feasibility in securing service discovery. Our primary focus was on technology that exists in layer 6 of the Open System Interconnection (OSI) reference model, the presentation layer. (3) Figure 1 shows each of the layers in the OSI model. The sections on the following pages touch upon technologies we researched as part of finding an appropriate solution.

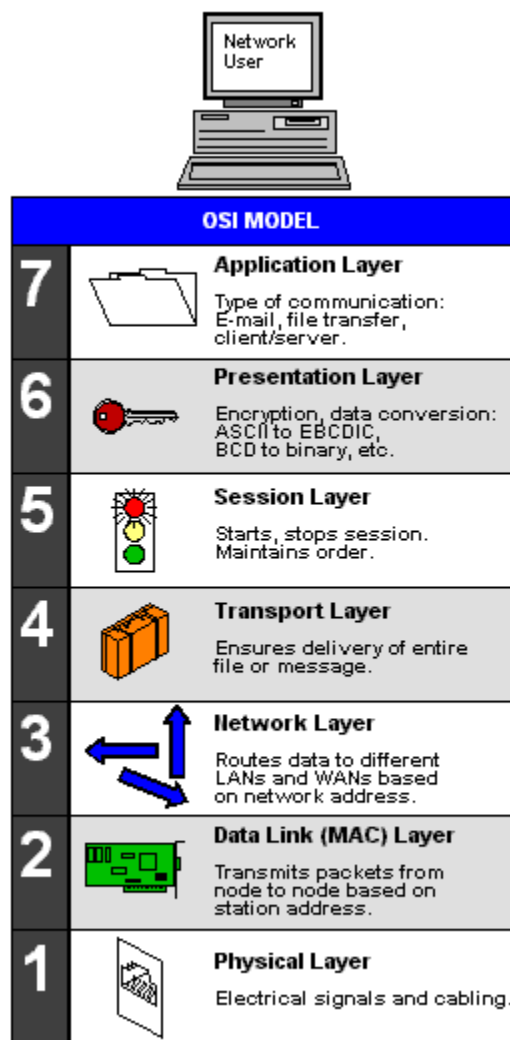


Figure 1 – The OSI model (3)

2.1 DHCP vs. link-local addressing

Each computer needs a unique address to function on a network. In the networking world this is known as an internet protocol (IP) address. An IP address is a unique 32-bit number assigned to each machine. A computer can either obtain an IP address from a central authority, such as a dynamic host configuration protocol (DHCP) server, or can assign its own address. If a DHCP server is not available, the computer can randomly select a number from a special pool of reserved, non-routable IP addresses for that purpose (169.254.0.0 – 169.254.255.255). The addresses from this pool are known as link-local addresses. (4) Some advances have been made to allow for multiple link-local subnets with routing across them. (5)

2.2 Multicast DNS

Multicast DNS is an implementation of a DNS server that listens and responds on a multicast address and port. The behavior of a multicast DNS server should be exactly the same as unicast DNS, in which a client can query any and all DNS servers on the multicast group and, if a match is found, a response is returned on that multicast group. The multicast address is defined to be 224.0.0.251, and the port is 5353. The internet draft proposal for multicast DNS states that any query for a DNS record with the .local suffix must be sent over multicast. (1) Queries and responses in multicast DNS are structured in the same way as unicast DNS. (4) The most common multicast DNS server is called the multicast DNS responder, which is known as mDNSResponder on Macintosh systems, as mDNSResponder.exe on Windows, and as mdnsd on Unix systems. (4)

To prevent excessive network traffic, multicast DNS typically follows some general best practices. First, responses are sent over multicast so that all nodes on the network become aware of the service. If there is no response to the initial query due to either dropped packets or no multicast DNS servers existing on the network, the DNS client will wait a period of time and then retransmit the query again over multicast. If there are multiple hosts on a network that all match a query, they will all respond. In future queries, a client can specify what is called a Known Answer List, meaning the client already knows about certain hosts and is looking for additional hosts. If a multicast DNS server sees a query and recognizes that it is already on the Known Answer List, it will not respond. This is common when a system is constantly polling the network at a certain interval (perhaps every 10 minutes) for new systems on the network. DNS responses also contain a time to live, which specifies how long the multicast DNS clients can treat the response as being valid in cache. In addition to probing the network through queries, multicast DNS also supports periodic announcements on the multicast group. The announcement is structured in the form of a DNS response, which allows all nodes on the network to update their local cache of information on the network. (4)

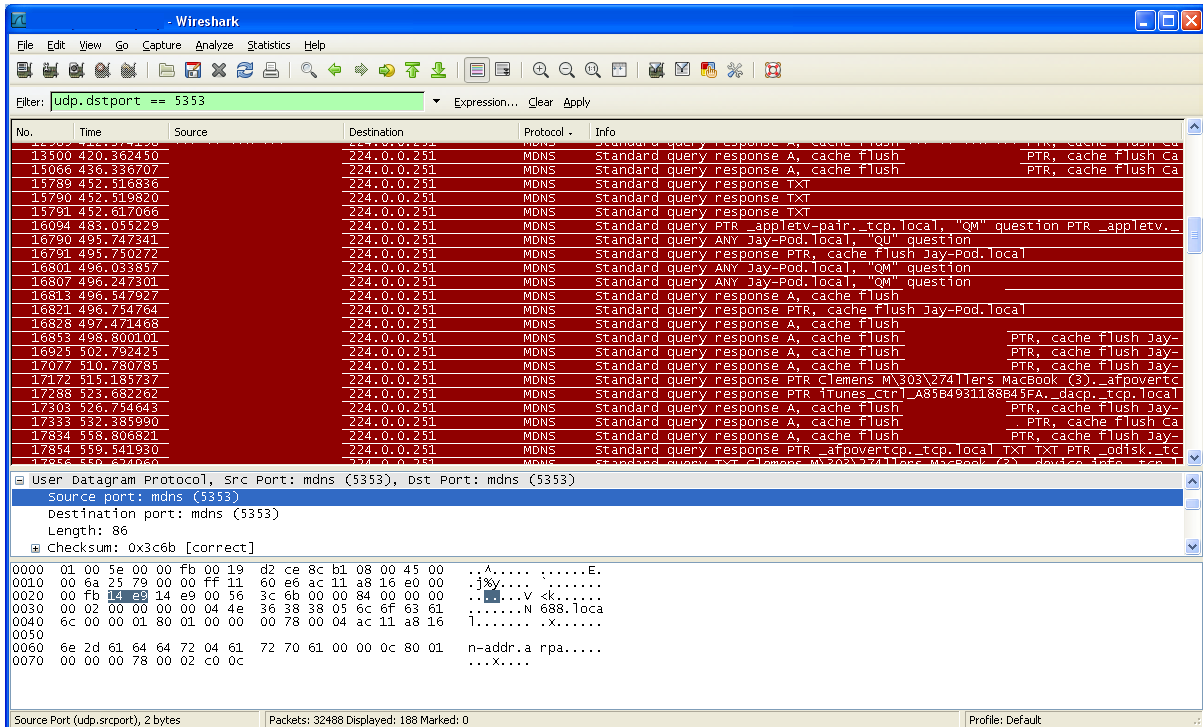


Figure 2 – Multicast DNS traffic while running iTunes™ on a public network

2.3 Secure DNS (DNSSEC)

DNS Security Extensions (DNSSEC) is “a suite of extensions that add security to the DNS protocol” (6). DNSSEC allows for a DNS domain to be signed with a key, certifying that the records are authentic and have not been tampered with. A top-level domain certifies the legitimacy of its child domains, which in turn sign for their children. DNSSEC uses new DNS record types such as DNSKEY, RRSIG, NSEC, and DS to support the integrity of its records. (7) DNSSEC is supported in both BIND (8) and Windows Server 2008 R2 (6). Not all top-level domains (such as .com and .net) are currently signed. Furthermore, the root zone is not signed. (9) Without these zones being signed, DNSSEC cannot be fully implemented.

2.3.1 DNSSEC lookaside validation

DNSSEC lookaside validation (DLV) is a short-term solution to providing secure DNS without having a signed root or parent zone. (9) (10) Domain lookaside validation works by implementing trusts to other domains in the form of DNS DLV records. For example, domains in the .com top-level domain can trust .org. Top-level domains are not the only domains that can implement DLV records. Any domain can implement a DLV to a top-level domain. (10) Another similar

temporary solution is to use trust anchors. Trust anchors are provided by the Internet Assigned Numbers Authority (IANA) for signing top-level domains only. From the trust anchors, the top-level domains can implement a DNSSEC hierarchy. Once the root zone is signed, the IANA trust anchor repository will cease to exist. (11)

2.3.2 Security concerns in DNSSEC

With the existence of NXT or NSEC records, it is possible for an outsider to discover the entire structure of the DNS domain, which can expose hidden records (such as contact information, public keys, and directory information). The counter argument, however, is by using DNSSEC, the information is freely available. The zone is signed so nobody can tamper with it. (12) When comparing DNSSEC to certificates in securing a DNS structure, EKRon (13) believes that DNSSEC is the better technology because internet-based certification authorities tend to be less secure.

2.4 PKI certificate DNS records

According to RFC 4398, DNS servers allow for the storage of certificate records, including PKIX, SPKI, and PGP records. These records are commonly known as a CERT resource record. The RFC allows for X.509 certificates that contain the user certificate, the certificate authority certificate, the authority revocation list, and the certificate revocation list. The RFC also allows for purpose-based certificates, such as TLS, S/MIME, and IPSec. However, if the DNS response is too big to fit in a UDP payload (for example, the certificate with all of the trusts contained within it), a URL with a link to the certificate could also be substituted, which is known as an indirect type DNS record. (12)

An example of a CERT record is the domain foo.com. In addition to having a CNAME record for www, there can be a CERT record in the domain that contains the SSL certificate for the www.foo.com website. Josefsson advocates the use of DNS as a certificate directory for a domain. (14)

2.5 Bonjour (DNS-SD) and Zeroconf

DNS Service Discovery is commonly known by its trademarked name of Bonjour. Service discovery is part of the larger initiative of zero configuration networking (zeroconf), a way for computers to configure themselves on a network without user intervention. Cheshire wrote that “DNS Service Discovery is a way of using standard DNS programming interfaces, servers, and packet formats to browse the network for services.” (15) RFC 2782 defines the DNS resource record for service entries (SRV). DNS service discovery leverages these SRV records to find a service on the network. SRV records follow the naming convention of *instance.servicetype.domain*, so an example of a printer using the internet printing protocol (IPP) type of service would have a service record of *some_printer_name.ipp.tcp.example.com*. (4) Since

service records behave the same way in both standard unicast DNS and multicast DNS, a SRV record for multicast DNS would look like *some_printer_name.ipp.tcp.local*. It is important to note that “DNS Service Discovery is compatible with, but not dependent on, Multicast DNS.” (15)

2.5.1 How query/response works in DNS-SD

Tools exist that allow for easy use of DNS-based service discovery. The most common system is DNS-SD.exe. To make a service discoverable, the service must be registered as a service. This is accomplished by running the register command (example `dns-sd -R test _daap._tcp . 1234`). At this point, there is a service named test that is available on TCP port 1234. Any system on a network that does a query for the test service on the multicast DNS port will get a response. (16)

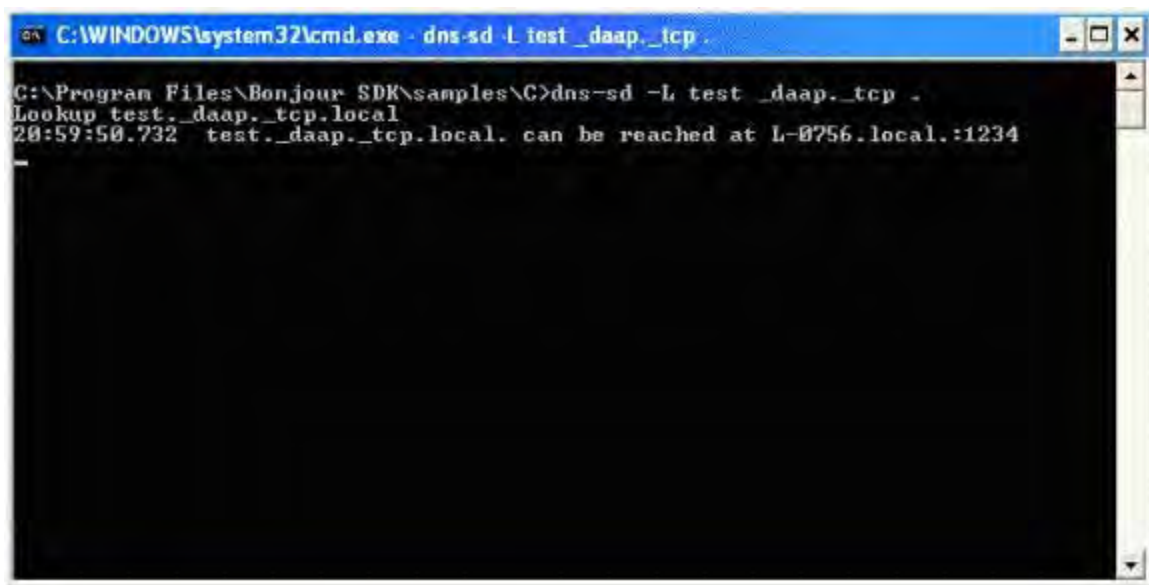


Figure 3 – Example of a DNS-SD query in Microsoft Windows XP (16)

2.6 Universal Plug and Play

Universal Plug and Play is another service discovery and zeroconf technology. It is based on ISO/IEC 29341. (17) Universal Plug and Play has been around since 1999 and has been implemented in versions of Microsoft Windows as early as Windows ME. Running on TCP/IP and HTTP, Universal Plug and Play implements the Simple Service Discovery Protocol to advertise itself on the network. The description of the service is written in XML. The XML contains URLs to control the device, manage device events, and pull presentation information from a device. (18)

2.7 Distributed registries

Popular with service oriented architectures, online registries exist where a user or system can advertise a service. Nodes on a network will have to become aware of an online central registry and use it every time a service is desired. This is analogous to a phone book that advertises services in specific categories (yellow pages). The burden is on the registry to make sure the services advertised in its registry are legitimate. Trabelsi's work on distributed registry models allows for secure access when storing new or existing records, and anonymous access for queries. (19) Trabelsi also goes on to ensure that these registries are resilient to attackers who wish to simply destroy the registry rather than take it over. (20)

2.8 Internet protocol version 6 (IPv6)

IP version 6 (IPv6) is a new protocol for internet addressing. It was designed by the Internet Engineering Task Force as a replacement for the current IPv4 standard of addressing. (21) RFC 2460 provides the specification for the protocol. Instead of using a 32-bit field to address nodes on the internet as in IPv4, IPv6 uses 128 bits, which allows for a larger number of addressable nodes. Furthermore, the IPv6 specification has greater support for multicast by adding a new scope field to the address. (22)

2.9 Secure IP (IPSec)

IPSec is a means of securing a network connection. IPSec was approved by the Internet Engineering Task Force to provide a point-to-point secure exchange of IP packets. IPSec can be deployed in transport or tunnel modes. (23) Transport mode provides a secure connection between two nodes by simply encrypting the payload. Tunnel mode encrypts and protects the entire IP packet by encrypting the entire message and wrapping it in a new IP packet. IPSec allows for virtual private networking (VPN) because the inner IP packet can be a private non-routable IP address. (24) IPSec is a mandatory part of IP version 6. (25)

There is also limited support for multicast in IPSec. Given that IPSec is designed for secure point-to-point communication, the notion of trying to facilitate key exchange among all members in the multicast group proves to be a challenge. This is especially the case when each node is both a producer and consumer and may join or leave a multicast group at any given moment. While authentication using IPsec may seem possible, encryption certainly is not. Most articles that are looking at multicast over IPSec are focusing on more traditional virtual private networking (VPN)-like uses of IPSec or Generic Routing Encapsulation GRE tunnels. (26)

3 Findings

Our original innovation grant proposal suggested that DNSSEC might be able to solve the problem of securing multicast DNS. Our research, however, showed that DNSSEC is inadequate and that DNS CERT records would be a better solution. Universal Plug and Play was not considered as a solution because it was too difficult to understand the API. That does not mean Universal Plug and Play is not a viable solution, but further research is necessary.

3.1 Scope of problem

In our research, we limited the scope of the problem to a local area network without a gateway to an external network. Time constraints limited the options space that could be explored. We made the following assumptions about this network to simplify the problem:

1. The network is running IPv4 only, without IPsec installed
2. Each node on the network has a unique IP address
3. Each machine has a unique PKI certificate issued to it by a certification authority
4. All machines on the network trust the issuing root certification authority
5. There is no central server on the network that is running a standard DNS server or acting as an authentication service
6. Nodes do not have a pre-existing direct trust between each other
7. No machines have been compromised by a hacker, virus, or other malicious activity

3.2 Why DNSSEC will not work

The first discovery in our research is that DNSSEC alone will not solve the problem because it requires a hierarchical trust structure (for foo.example.com to be secure, example.com must be secure). For the scenario to work, the .local domain will need to be secured first, but this cannot happen because there is no central DNS repository for .local. (4) Also, the computers cannot implement domain lookaside validation, because to do so two computers would have to trust each other, which violates assumption 6 in section 3.1.

3.3 CERT records as a solution

An alternative to DNSSEC would be to include CERT records in the multicast DNS server. For example, suppose a computer queries for a service and gets a response from another computer. The node doing the querying can then request a CERT record from the host with the service. If the

host responds with a certificate, the querying computer can then validate the certificate against its repository of trusted root certification authorities.

This alternative is backwards compatible with existing infrastructure – if a legacy system queries for a service, it will still get a multicast DNS response, but it won't ask for a CERT record. Likewise, if a newer system queries for a service on a legacy system, it will get a service response but no CERT record. The burden is then on the querying node to determine if the legacy system is trustworthy.

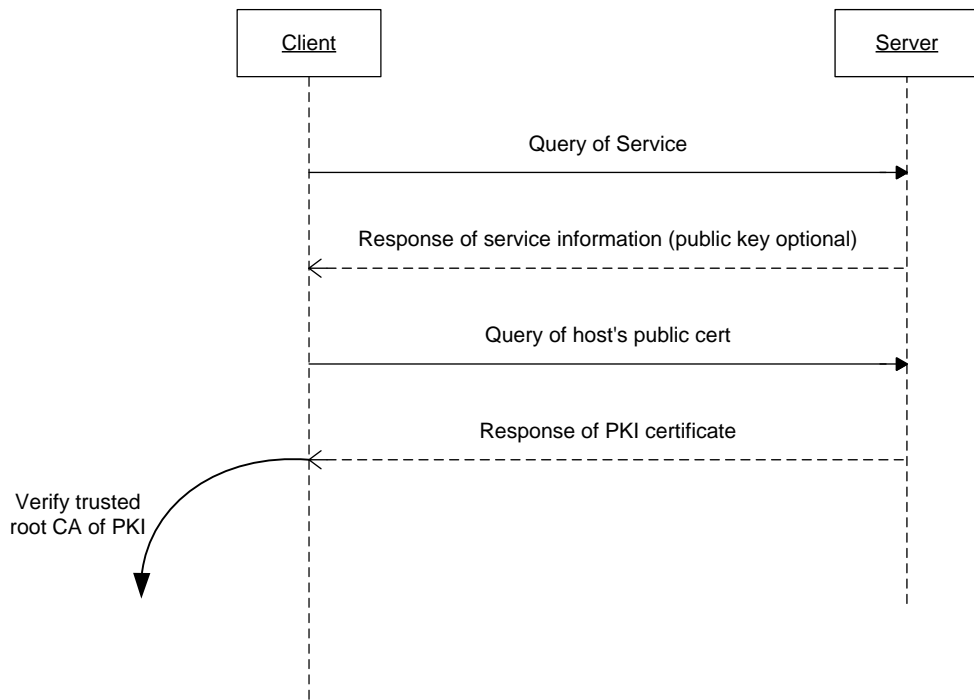


Figure 4 – Client/server interaction in service discovery with certificates

3.3.1 Trusted root certification authorities

PKI certificates come in many flavors. Certificates can be issued from a known and reputable source, such as VeriSign, or they can be issued locally by the computer. These reputable sources

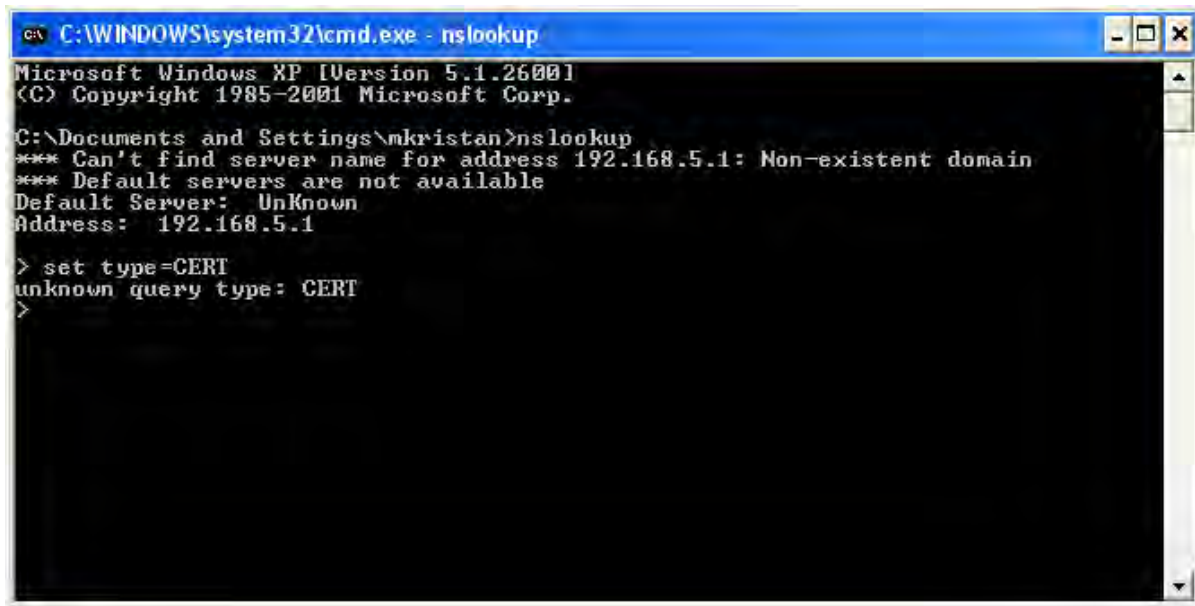
are known as Certification Authorities (CAs). For the CERT record solution to work, the certificates used by all of the machines in the service discovery need to have a common root CA, or at least know about (and trust) the other CAs. Furthermore, the certification authority does not need to be a globally known commercial entity. If someone is looking to only share a service among a close group of machines in which one person/company has direct control over all systems, then standing up a local CA and issuing certificates to all nodes will be sufficient. Keep in mind, however, that external nodes will not know about the CA and will complain about the validity of the certificates.

3.3.2 User certificates vs. machine certificates

Certificates can be issued to an individual user or to a local machine. There are tradeoffs to using either in the realm of service discovery. In the machine model, it is very easy to query for a host's DNS CERT record by simply knowing the hostname or finding it out via a reverse IP lookup. Trying to find the correct user's certificate in the DNS query is difficult because it is not clear in service discovery which user owns the service. It is important that all services that run on the host use the same certificate. If not, there will be a mismatch between the DNS lookup of the certificate and the service itself. In the user model, the user is the one starting up the service or trying to connect to the service. Therefore, it makes sense to verify that the user is authorized to access the service since many people can use the same computer. The conclusion is that the user's need for access should be determined by a secondary authentication model after the initial service discovery. Two examples are a simple logon or key exchange.

3.4 Support in current technology

In looking at currently available systems, we found that neither BIND, an open source DNS server, nor Windows Server appear to support CERT records. It is also unclear if multicast DNS has CERT support. According to Cheshire's DNS-based service discovery document (on which Bonjour is based), it does not appear that support exists there either. The only supported types used in multicast DNS are A, PTR, TXT, and SRV records. (27) It is possible in theory to configure a traditional DNS server to listen on a multicast address, although there is no evidence of that being encouraged or supported. An alternative is to develop a new multicast DNS responder application with that support natively available, since multicast API libraries do exist. (28)



```
C:\WINDOWS\system32\cmd.exe - nslookup
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\mkristan>nslookup
*** Can't find server name for address 192.168.5.1: Non-existent domain
*** Default servers are not available
Default Server: Unknown
Address: 192.168.5.1

> set type=CERT
unknown query type: CERT
>
```

Figure 5 – Windows XP does not support CERT records in nslookup

4 Conclusion

In conclusion, it is feasible to secure service discovery through multicast DNS. The findings in section 3 represent a possible solution to the problem area of trusted service discovery. However, it is by no means the one and only solution. The DNS certificate solution proposed in this document will not make the system ultimately secure. Instead it is merely one piece in the complex security puzzle. One example not covered in this document is ensuring that the computer itself is not compromised. Even if the client knows that a computer offering up services was at one point trusted, there is no guarantee it will remain secure.

4.1 Next steps

The next step in this research is to build a prototype system that can discover a service and then query for its trusted identity. The prototype should only connect to services that have valid identities. To maintain backwards compatibility, the system should also interact seamlessly with an existing set of clients and services that do not have DNS CERT records implemented. The desired behavior is for a gracious ignore from the DNS server, which would then prompt the querying client to prompt the user on what to do next (proceed without the CERT record or do not connect). Once this prototype is complete and a proof of concept is developed, steps should be taken to remove some of the assumptions stated in section 3.1 to further generalize the problem.

Beyond the scope of secure service discovery is the idea of zero configuration networking (zeroconf). A lot of work will be needed to make sure these new systems are easy to use. Consultation with human factors experts and software developers will be needed to make the process of issuing certificates, discovering services, and verifying trust as seamless as possible. Research should include technologies that companies and users currently use to process authentication, especially in the realm of PKI. Further research is necessary to either find a more suitable alternative, or make PKI easier to deploy in an ad hoc environment for the purpose of service discovery. Having a PKI-based solution assumes that one will invest the time and effort in developing a complete infrastructure for issuing and validating certificates.

References and Bibliography

1. **Ceshire, Stuart and Krochmal, Marc.** Multicast DNS. *multicastdns.org*. [Online] Apple Inc., September 10, 2008. [Cited: September 20, 2009.] IETF Internet Draft. <http://files.multicastdns.org/draft-cheshire-dnsext-multicastdns.txt>.
2. **Huston, Geoff.** DNSSEC - The Theory. *The ISP Column*. [Online] Geoff Huston, August 2006. [Cited: August 24, 2009.] <http://www.potaroo.net/papers/isoc/2006-08/dnssec.html>.
3. **ZDNet.** OSI Model: Definition and additional resources from ZDNet. *ZDNet*. [Online] 2009. [Cited: September 20, 2009.] <http://dictionary.zdnet.com/definition/OSI+model.html>.
4. **Cheshire, Stuart and Steinberg, Daniel H.** *Zero Configuration Networking: The Definitive Guide*. 1st Edition. Sebastopol : O'Reilly Books, 2006. ISBN 0-596-10100-7.
5. *An IP Address Configuration Algorithm for Multi-Router Zeroconf Networks.* **Akinlar, Cuneyt, et al.** s.l. : IEEE, 2002. Proceedings of the Seventh International Symposium on Computer and Communications (ISCC'02). 1530-1346.
6. **Microsoft Corporation.** Domain Name System Security Extensions. *Microsoft*. [Online] 2009. [Cited: September 20, 2009.] <http://www.microsoft.com/downloads/details.aspx?FamilyID=7A005A14-F740-4689-8C43-9952B5C3D36F&displaylang=en>.
7. **Clegg, Alan.** DNSSEC in 6 minutes. *Internet Systems Consortium*. [Online] 1.5, 2008. [Cited: September 20, 2009.] Powerpoint presentation in PDF form. https://www.isc.org/files/DNSSEC_in_6_minutes.pdf.
8. **Albitz, Paul and Liu, Cricket.** *DNS and Bind*. 4th Edition. Sebastopol : O'Reilly Books, 2001. ISBN 0-596-00158-4.
9. **Afilias.** ISC, Afilias and Neustar Bring Secure DNS One Step Closer. *Afilias*. [Online] Afilias, August 10, 2009. [Cited: September 27, 2009.] <http://www.afilias.info/news/2009/08/05/isc-afilias-and-neustar-bring-secure-dns-one-step-closer>.
10. **Weiler, S.** DNSSEC Lookaside Validation (DLV). *DNSSEC Lookaside Validation (DLV)*. [Online] The Internet Engineering Task Force (IETF), November 2007. [Cited: September 27, 2009.] <http://tools.ietf.org/html/rfc5074>. RFC 5074.
11. **IANA.** Interim Trust Anchor Repository. *Internet Assigned Numbers Authority*. [Online] Internet Corporation for Assigned Names and Numbers. [Cited: September 27, 2009.] <https://itar.iana.org/>.

12. **Josefsson, S.** Storing Certificates in the Domain Name System (DNS). *IETF*. [Online] March 2006. [Cited: August 24, 2009.] <http://tools.ietf.org/html/rfc4398>. RFC 4398.
13. **EKRon.** DNSSEC versus certificates. *Educated Guesswork*. [Online] January 18, 2009. [Cited: August 24, 2009.] http://www.educatedguesswork.org/2009/01/dnssec_versus_certificates.html.
14. **Josefsson, Simon.** LDAP and DNS as Certificate Directories. *Simon Josefsson's Master's Thesis*. [Online] Simon Josefsson, January 7, 2002. [Cited: September 27, 2009.] <http://josefsson.org/master-thesis/node9.html>.
15. **Ceshire, Stuart.** DNS Service Discovery (DNS-SD). [Online] 2009. [Cited: August 18, 2009.] <http://www.dns-sd.org>.
16. DNS Based Service Discovery (DNSSD). *Smallegan*. [Online] June 9, 2005. [Cited: September 27, 2009.] <http://www.smallegan.com/blog/images/dnssd/ss4.jpg>.
17. **International Organization for Standardization.** Information technology -- UPnP Device Architecture -- Part 1: UPnP Device Architecture Version 1.0. *ISO*. [Online] 2009. [Cited: September 27, 2009.] http://www.iso.org/iso/catalogue_detail?csnumber=52674.
18. **Microsoft Corporation.** Understanding Universal Plug and Play. *UPnP Forum*. [Online] June 2000. [Cited: September 27, 2007.] http://www.upnp.org/download/UPNP_UnderstandingUPNP.doc.
19. *Secure Service Discover with Distributed Registries*. **Trabelsi, Slim and Roudier, Yves.** s.l. : IEEE, 2008, pp. 1-6. 978-1-4244-3062-8.
20. *On the Impact of DoS Attacks on Secure Service Discovery*. **Trabelsi, Slim, Guillaume, Urvoy-Keller and Roudier, Yves.** s.l. : IEEE, 2008, pp. 532-537. 978-7695-3492-3.
21. IPv6: The Next Generation Internet. [Online] IPv6 Information Page, April 14, 2003. [Cited: September 27, 2009.] <http://www.ipv6.org/>.
22. **Deering, S and Hinden, R.** Internet Protocol, Version 6 (IPv6) Specification. *USC Information Sciences Institute*. [Online] Network Working Group, December 1998. [Cited: September 27, 2009.] <ftp://ftp.isi.edu/in-notes/rfc2460.txt>. RFC 2460.
23. **Webopedia.** What is IPsec? *Webopedia*. [Online] WebMediaBrands Inc., May 20, 2004. [Cited: September 27, 2009.] <http://www.webopedia.com/TERM/I/IPsec.html>.
24. **Friedl, Steve.** An Illustrated Guide to IPsec. *Unixwiz.net - Software Consulting Central*. [Online] August 24, 2005. [Cited: September 27, 2009.] <http://unixwiz.net/techtips/iguide-ipsec.html>.

25. **Das, Kaushik.** IPsec & IPv6 - Securing the NextGen Internet. *IPv6*. [Online] IPv6 Inc., 2008. [Cited: September 27, 2009.] <http://www.ipv6.com/articles/security/IPsec.htm>.
26. **McKeag, Louise.** Adding multicast to IPsec. *TechWorld*. [Online] April 20, 2004. [Cited: September 20, 2009.] <http://howto.techworld.com/security/511/adding-multicast-to-ipsec/>.
27. **Cheshire, Stuart and Krochmal, Marc.** DNS-Based Service Discovery. *DNS-SD Website*. [Online] Apple Inc., March 10, 2009. [Cited: September 27, 2009.] <http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt>. Internet Draft.
28. **Jepri and Jer.** Net::MDNS::Server - Perl extension for a multicast DNS server. *Comprehensive Perl Archive Network*. [Online] Finnish University and Research Network, 2003. [Cited: September 20, 2009.] <http://kobesearch.cpan.org/htdocs/Net-MDNS-Server/Net/MDNS/Server.html>.
29. *Autoconfiguration for IP Networking: Enabling Local Communication.* **E.Guttman.** 2001, IEEE Internet Computing, pp. 81-86.
30. *Accelerating Service Discovery in Ad-hoc Zero Configuration Networking.* **S.G. Hong, S. Srinivasan, H. Schulzrinne.** 2007, Proceeding from IEEE GLOBECOM, pp. 961-965.
31. *z2z: Discovering Zeroconf Services Beyond Local Link.* **J.W. Lee, H. Schulzrinne, W. Kellerer, Z. Despotovic.** 2007, IEEE Globecom Workshops, pp. 1-7.
32. *Facilitating Secure Ad hoc Service Discovery in Public Environments.* **F. Zhu, M. Mutka, L. Ni.** Proceedings of the 27th IEEE Annual International Computer Software and Applications Conference, pp. 1-6.
33. *Secure Service Discovery in Home Networks.* **Scholten, Hans, et al.** s.l. : IEEE, 2006, pp. 115-116.
34. *Private and Secure Service Discovery via Progressive and Probabilistic Exposure.* **Zhu, Feng, et al.** 11, s.l. : IEEE Computer Society, 2007, IEEE Transactions on Parallel and Distributed Systems, Vol. 18, pp. 1565-1577. 1045-9219.
35. **Unknown.** List of DNS record types. *Wikipedia*. [Online] August 14, 2009. [Cited: August 24, 2009.] Not a primary source of information. http://en.wikipedia.org/wiki/List_of_DNS_record_types.
36. **Petri, Daniel.** Understand GlobalNames Zone in Windows Server 2008. *Petri IT Knowledge Base*. [Online] Blue Whale Web Inc., January 7, 2009. [Cited: August 20, 2009.] <http://www.petri.co.il/windows-DNS-globalnames-zone.htm>.

37. **Anteniese, Dr. Giuseppe and Danilov, Claudiu.** Integrating OpenSSH with Secure DNS. *John Hopkins University Department of Computer Science*. [Online] March 12, 2001. [Cited: August 24, 2009.] <http://www.cs.jhu.edu/~claudiu/projects/dnssecssh.html>.

38. IPv6. *Wikipedia*. [Online] September 20, 2009. [Cited: September 20, 2009.] <http://en.wikipedia.org/wiki/IPv6>.

39. **Komar, Brian.** *Windows Server 2008 PKI and Certificate Security*. [ed.] Denise Bankaitis, Karen Szall and Martin DelRe. 1st Edition. Redmond : Microsoft Press, 2008. ISBN-13: 978-0-7356-2516-7.

Acknowledgements

I would like to acknowledge Dr. Sandeep Mulgund for encouraging me to apply for an innovation grant, which has allowed me to perform this research. I would also like to acknowledge my department head Douglas Robbins for his support and suggestions. Finally, I would like to thank Dr. Christopher Niessen, Jonathan Wielicki, Charles Holmes, and Erin Connors for providing suggestions on my proposal and feedback on a draft of this report.

About the author

Michael Kristan is a Senior Software Systems Engineer in department E146 (Air Force Research and Edge Innovation) at The MITRE Corporation. Prior to joining MITRE, he worked in various high tech firms around the Massachusetts area. He holds a B.S. in both Computer Science and Electrical & Computer Engineering from Worcester Polytechnic Institute. He also holds a graduate certificate in Software Engineering. Interest areas include agile software development practices, enterprise IT architecture, and network administration.

MITRE Innovation Program

This technical report was generated from a division-funded innovation grant in the 2009 fiscal year. The MITRE Innovation Program (MIP) seeks to address the nation's most pressing challenges through open technological and scientific innovation. The MIP complements MITRE's direct work program by striving to see these challenges in fundamentally new ways, thereby inspiring innovative – and sometimes radical – solutions. The program promotes research and development to advance and apply emerging technologies to meet sponsor needs.

To ensure that knowledge gained through the program becomes widely available, MIP researchers publicize their findings through conference presentations, journal papers, consortia and standards bodies, along with events such as the annual Innovation Exchange. In the course of seeking and applying this knowledge, we often create intellectual property that is useful to our government sponsors and to the public.

<http://www.mitre.org/work/program.html>

List of acronyms

API	Application Program Interface
BIND	Berkeley Internet Name Domain
CA	Certification Authorities
CERT	Certificate
CNAME	Canonical Name
DHCP	Dynamic Host Configuration Protocol
DLV	DNSSEC Lookaside Validation
DNS	Domain Name System
DNS-SD	Domain Name System Service Discovery
DNSKEY	Domain Name System Key
DNSSD	Domain Name System Service Discovery
DNSSEC	Domain Name System Security
DoS	Denial of Service
DS	Data Segment
GRE	Generic Routing Encapsulation
HTTP	Hyper Text Transfer Protocol
IANA	Internet Assigned Numbers Authority
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPP	Internet Printing Protocol
IPSec	Internet Protocol Security
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
MDNS	Multicast Domain Name System
MIME	Multipurpose Internet Mail Extension
MIP	MITRE Innovation Program
NSEC	Next Secure
NXT	Next
OSI	Open System Interconnection
PDF	Package Definition File
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure X.509
PTR	Pointer
RFC	Request For Comment
RRSIG	Resource Record

SD	Service Discovery
SPKI	Secure Public Key Infrastructure
SRV	Service
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
TXT	Text
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
VPN	Virtual Private Networking
XML	Extended Markup Language

Distribution List

E123

D. Holtzman

E140

M. Heller

T. Szczerbinski

E142

R. Miller

E143

J. Dominick

L. Fleming

B. Steele

E146

R. Bahnij

E. Connors

L. DeYoung

P. Gonsalves

C. Holmes

J. Jacoby

M. Kristan

M. Los

S. Mulgund

C. Quigley

D. Robbins

S. Sibberson

S. Wiebenson

J. Wielicki

E302

T. Jensen

E305

P. Chen

A. Lastra

G. Stafford

E536

C. Niessen

M. Zimmermann

G132

P. Lucia

R301

W. Swirnoff

Z021

L. Barrett