Replacing WINS in an Open Environment with Policy Managed DNS Servers

Mark Lucas

Replacing WINS in an Open Environment with Policy Managed DNS Servers

GIAC (GCWN) Gold Certification

Author: Mark J. Lucas, mark.lucas@student.sans.edu
Advisor: Tanya Baccam

Accepted: August 5, 2020

Abstract

In some environments, Windows workstations require placement on the open internet. In order to protect the read-write domain controllers, administrators locate them in a protected enclave behind a firewall, and read-only domain controllers authenticate workstations during day-to-day operations. While this is strong protection for the read-write domain controllers, the configuration breaks the standard dynamic DNS registration of Windows workstations with the read-write domain controller. In our environment, we have maintained WINS servers linked to Windows DNS via the WINS lookup function to continue finding workstations by name. The TechNet page on WINS (Davies, 2011) was last updated almost nine years ago, and Microsoft has been actively encouraging the abandonment of WINS (Ross & Mcillece, 2020). This paper explores Windows DNS Policies to replacing WINS with Dynamic DNS and policy-controlled responses to queries. Utilizing source IP addresses, DNS policies can regulate the provided answers. The operability of DNS Policies and the applicability to this solution is evaluated in depth.

1. Introduction

Many corporations and institutions are in protected enclaves where internal networks are behind firewalls, and external traffic is virtually non-existent. In these environments, the use of "split-brain" DNS with two different servers is viable (Ross & et al., 2020d). However, there are environments where devices require the open internet and yet interact with protected Active Directory servers for management and authentication. We will examine one such educational environment. The feasibility of Windows DNS Policies to protect private DNS records while accurately answering queries from the open internet will be explored.

1.1. History

The institution embraces an open environment where collaboration, whether internally or worldwide, is paramount. To support this goal, the institution network sits on the public internet with few restrictions on the border router. All workstations and some servers have publicly routable addresses allowing for communications worldwide despite the essential recommendation from the Center for Internet Security to utilize firewalls (Center for Internet Security, 2019). Security is a consideration at the institution; therefore, many servers, including the authentication infrastructure, are isolated behind server room level firewalls.

1.1.1. Early History

Windows name resolution at the institution has been an ongoing challenge. Deploying WINS on campus for workstation name resolution began circa 1997 with Windows NT 4.0. The primary DNS infrastructure is BIND-based and manually updated with each static entry. The Windows Active Directory Domain is a delegated sub-domain so that the complicated Active Directory SRV records did not have to be manually created and maintained in BIND. Institution DHCP is Linux-based and configured with the BIND DNS servers and the WINS servers. WINS provides a non-invasive way for Windows System Administrators and Help Desk personnel to discover and connect to any device on campus.

Mark J. Lucas, mark.lucas@student.sans.org

When rolling out Windows 2000 Active Directory, the institution took advantage of the WINS lookup feature within Windows DNS. This DNS feature supplemented the new Dynamic DNS registration for domain-joined workstations and maintained all devices' visibility on campus. While there were several firewalls on campus to protect application and web server functionality, administrators placed read-write domain controllers on the open internet with operating system firewalls rules which restricted access to the institution IP address space. This placement, along with IP address re-write functionality on the firewalls, facilitated correct name resolution inside and outside the firewalls. This domain controller configuration was an effective solution for several years until a security incident forced the implementation of read-only domain controllers on the public network with read-write domain controllers isolated behind firewalls.

1.1.2. Challenges

The Active Directory sub-domain contained records that had to be available publicly, such as MX and web server addresses. It was unwise to open the read-only domain controllers to the public; so the team created secondary, stand-alone Windows DNS servers to handle the records. To reduce the workload on System Administrators, the secondary DNS servers were made replicas of the read-write domain controllers records.

Difficulties quickly arose due to the advertisement of private IP addresses from inside the firewall on the public internet. Administrators used various techniques and workarounds until it became apparent a true split-brain scenario was required. Converting the secondary servers to primary servers and manually entering dual records as necessary solved the issue.

More disturbing was that these changes completely blocked workstations' ability to dynamically and securely register their name with Windows DNS. WINS, a backup for non-domain devices, was now the only option other than manual entry. Manual entry was untenable due to the use of DHCP for most of the 2,500 managed workstations. Therefore, WINS was the only choice until the advent of DNS Policy in Windows Server 2016.

1.2. WINS vs. DNS

1.2.1. WINS History

WINS was initially designed for small workgroup networks, based on NetBIOS (Network Basic Input/Output System), and created by Sytek in 1983 (Mitchell, 2020). Originally, all name registration and discovery were broadcast-based so that every machine on the network received the name announcement and query. The namespace was flat such that a specific name might only be used once in each network segment. Broadcast messages also did not cross router boundaries, limiting name resolution to the local network space.

The Microsoft implementation of WINS uses 15 characters for the resource name and the 16th character to indicate the resource type. Microsoft WINS introduced WINS servers, which replicate the WINS name databases, providing extended name resolution. This new implementation also provided for reduced broadcast messages by registering names directly with the WINS server. Any network segment could query the WINS server; eliminating the router boundary. (Microsoft Corporation, 1996).

1.2.2. DNS History

At about the same time WINS was created, DNS (Domain Name System) was also developed and adopted as an internet standard in 1986. Paul Mockapetris created the system that permitted names to be used rather than IP addresses. The need for the system became apparent as the multiple individual networks throughout the world were becoming connected to what was to become known as the "internet." (Pramatarov, 2018). RFC 1035 outlines the hierarchical nature of domain names allowing for wholly unique names across many different systems. With this innovation, the resolution of any name to a unique IP address became possible. (Mockapetris, 1987).

DNS greatly expanded the length of the resource name from WINS 15 characters (plus one character for resource type) to 255 characters for the full resource name. Additional information was added to the request and answer to include the type of record, class, and time to live (TTL). Time to live allowed for the automatic expiration of the record so that stale information was not propagated (Mockapetris, 1987).

1.2.3. Comparison

In 1998, The Microsoft TechNet library provided a convenient table outlining some of the differences between WINS and DNS (Masterson, Knief, Vinick, & Roul, 1998).

WINS	DNS
The purpose is to resolve NetBIOS names to IP addresses.	The purpose is to resolve host names to IP addresses.
Names are flat and 15 characters long.	Names are hierarchical in nature.
Name registration is dynamic and happens automatically.	Name registration is static and has to be done manually.
Supports incremental replication of the data, which means that only changes in the database are replicated between WINS servers.	Doesn't support incremental replication of data between DNS servers. This means the whole database has to be replicated every time.
Supports DHCP.	Doesn't support DHCP.
Doesn't support email routing or additional TCP/IP application services.	Supports other TCP/IP application services such as email routing.

Table 1 - WINS Verses DNS

Since 1998, many changes occurred to Windows DNS protocols. These changes eliminated some of the negatives on the DNS side by adding functionality available in WINS. Most importantly, Windows 2000 introduced dynamic registration based on Windows domain membership and incremental replication of domain name data through Active Directory integration of the DNS databases.

The most recent changes introduced in Windows Server 2016 include functionality to address the problems presented in this paper. Three of the new functions presented in the Microsoft article "What's New in DNS Server in Windows Server" explored are:

- **Geo-Location Based Traffic Management.** You can use DNS Policy to allow primary and secondary DNS servers to respond to DNS client queries based on the

geographical location of both the client and the resource to which the client is attempting to connect, providing the client with the IP address of the closest resource.

- **Split-Brain DNS.** With split-brain DNS, DNS records are split into different Zone Scopes on the same DNS server, and DNS clients receive a response based on whether the clients are internal or external clients. You can configure split-brain DNS for Active Directory integrated zones or for zones on standalone DNS servers.

- **Filtering.** You can configure DNS policy to create query filters that are based on criteria that you supply. Query filters in DNS policy allow you to configure the DNS server to respond in a custom manner based on the DNS query and DNS client that sends the DNS query (Ross, McIllece, Poggemeyer, & Yisheng, 2020). These three features provide the basis for solutions to the problems presented. Regulating the answers provided by the DNS server to queries originating from different IP address ranges will allow for a single Active Directory-based DNS database to service all queries with the correct and secure answers. These features also allow for the configuration of recursive vs. non-recursive queries based upon the source of the request.

2. Environmental Requirements

In many corporate and institutional situations, the use of a perimeter firewall, which provides a protected enclave for all workstations and servers, allows for a flat DNS structure. Even with additional firewalls safeguarding the servers from the workstations, a flat DNS structure internally is entirely possible with the implementation of firewall features similar to Cisco's ASA firewall feature of DNS Inspection. The DNS Inspection feature allows for the translation of DNS IP addresses based on the Network Address Translation (NAT) configuration (Cisco Systems Inc., 2018). Typically, in these situations, the internal domain is separate from the public face of the organization. This separation makes the DNS record challenges minimal, and a separate server or cloud-based provider handles DNS records.

Other situations place workstations on the public internet while some servers are behind the firewalls. The DNS structure is flatter and the public and private DNS records are intermingled, introducing challenges to manage what records are available to the public vs. internal devices. In these situations, Active Directory domain records, internal institution servers, application, and website names and addresses should not be publicly discoverable. While "security through obscurity" is not a panacea against exploitation by malicious entities, controlling the flow of information out of the organization is a primary tenant of security. CIS Control 20 encourages organizations to "Include tests for the presence of unprotected system information and artifacts that would be useful to attackers, including network diagrams, configuration files, older penetration test reports, e-mails or documents containing passwords or other information critical to system operation." (Center for Internet Security, 2019). DNS records of workstations, private servers, and network appliances allow an attacker to easily construct a network map of the organization and discover weaknesses and interesting private caches of data.

Therefore, it is in the best interest of organizations to protect internal DNS records and only expose those records necessary for the functionality of publicly facing applications.

3. Research Method

3.1. Environment Setup

A test environment was created to explore the new policy features and determine the suitability of the policies to solve the issues presented. VMware Fusion was used to create three virtual machines:

- Read-write domain controller, Windows 2016 Standard
- Read-only domain controller, Windows 2016 Standard
- Client-server, Windows 2016 Standard

The domain controller servers were configured with the default DNS configuration in the domain "dragon.apollo." Additional DNS records were added for testing purposes. These additional records are:

- A – rwdc.dragon.apollo = 192.168.111.69
- A – rwdc.dragon.apollo = 192.168.111.169
- A – web.dragon.apollo = 192.168.111.68
- A – web.dragon.apollo = 192.168.111.168
- A – rodc.dragon.apollo = 192.169.111.67
- A – rodc.dragon.apollo = 192.169.222.167
- CName – ReadWriteDC.dragon.apollo = rwdc.dragon.apollo
- CName – ReadOnlyDC.dragon.apollo = rodc.dragon.apollo
- CName – web.dragon.apollo = lunar.dragon.apollo
- MX – mail.dragon.apollo = 192.168.111.192

192.168.111.0/25 is considered the private network. 192.168.111.128/25 is considered the public network.

The following network protocols are turned off to remove discovery mechanisms or other avenues of network connectivity other than IPv4.

- Microsoft Network Adapter Multiplexor Protocol
- Microsoft LLDP Protocol Driver
- Internet Protocol Version 6 (TCP/IPv6)
- Link-Layer Topology Discovery Responder
- Link-Layer Topology Discovery Mapper I/O Driver

While the DNS Policy features support IPv6, this test is limited to IPv4. To reduce the complexity of this test, with Windows Defender Firewall with Advanced Security is turned off. In a real deployment, the firewall should be deployed and configured to permit appropriate traffic.

To simulate DNS queries from the public network vs. the private network, the IP address of the webserver was changed from 192.168.111.67 to 192.169.111.167.

All three Windows servers had access to the internet to obtain the latest patches and updates from Microsoft; however, no traffic was permitted from the Host machine nor the internet into the private VMware network of these machines. All machines were fully patched.

VMware Tools were installed on all three machines to facilitate the management of the VMs. Notepad++ was installed on all three computers to facilitate the reading of logs and modifications to DNS files.

IIS was installed on the webserver. Active Directory Domain and DNS services were installed on the two domain controllers. The RODC and RWDC servers were joined to the domain "dragon.apollo." The webserver was left as stand-alone for testing purposes.

IP Address Management (IPAM) (Ross & et al., 2020b) was installed and later removed on the webserver. There was an indication in the Microsoft document "Set Access Scope for a DNS Zone" (Ross & et al., 2020c) that the IPAM tool would be useful in managing the DNS Scopes. Instead, IPAM can view the existing DHCP and DNS zones, and control how DHCP records IP addresses in the zones. Management of the DNS scopes, policies associated with the scopes, and manually managed IP addresses are not supported. Research stopped upon discovery that IPAM was not helpful. The availability of DNS policy and scope management only exists in PowerShell.

Creation of the testing scenarios below was accomplished using PowerShell syntax documented in "DNS Policy Scenario Guide" (Ross & et al., 2020a), "Use DNS Policy for Split-Brain DNS in Active Directory" (Ross & et al., 2020d), and "DnsServer" (Microsoft Corporation, 2020). DnsServer is the master page for all PowerShell commands related to the management of DNS servers. Various sub-pages for specific commands were referenced from this master page.

3.2. Tests

3.2.1. Recursion

Selectively removing the ability for DNS servers to provide recursive answers is a longstanding best practice (Infoblox, 2018). This simple configuration eliminates the ability of attackers to perform DNS Amplification attacks.

Line 1 disables all recursion.

Line 2 enables recursion for internal clients.

Line 3 defines the subnet permitted for recursive queries.

Line 4 creates the policy using the elements from lines 2 and 3.

Line 5 restarts the DNS service, a requirement for the changes to take effect.

```
Set-DnsServerRecursionScope -Name . -EnableRecursion $False
Add-DnsServerRecursionScope -Name "InternalClients" -EnableRecursion $True
Add-DnsServerClientSubnet -Name "RecursionSubnet" -IPv4Subnet 192.168.111.0/25
Add-DnsServerQueryResolutionPolicy -Name "SplitBrainRecursionPolicy" -Action ALLOW -App
lyOnRecursion -RecursionScope "InternalClients" -ClientSubnet “eq,RecursionSubnet”
restart-service DNS -passthru
```

3.2.2. Managing answers to resource record queries

A very useful feature of DNS policies is to control what answers are provided to different source IP addresses. With this feature, split brain functionality can be handled by a single DNS database. Also, internal records can be completely hidden from public queries.

Lines 1-4 define the subnets.

Lines 5-6 create the new scopes within the root zone.

Lines 7-8 create the resource records within a new (or existing) zone scope.

Lines 9-10 use the above three elements to create the policy defining the answer provided to each subnet.

```
Add-DnsServerClientSubnet -Name "PrivateSubnet1" -IPv4Subnet "192.168.111.0/26"
Add-DnsServerClientSubnet -Name "PrivateSubnet2" -IPv4Subnet "192.168.111.64/26"
```

```
Add-DnsServerClientSubnet -Name "PublicSubnet1" -IPv4Subnet "192.168.111.128/26"

Add-DnsServerClientSubnet -Name "PublicSubnet2" -IPv4Subnet "192.168.111.192/26"

Add-DnsServerZoneScope -ZoneName "Dragon.Apollo" -Name "InstitutePrivateScope"

Add-DnsServerZoneScope -ZoneName "Dragon.Apollo" -Name "InstitutePublicScope"

Add-DnsServerResourceRecord -ZoneName "Dragon.Apollo" -A -Name "www" -IPv4Address "192.
168.111.68" -ZoneScope "InstitutePrivateScope"

Add-DnsServerResourceRecord -ZoneName "Dragon.Apollo" -A -Name "www" -IPv4Address "192.
168.111.168" -ZoneScope "InstitutePublicScope"

Add-DnsServerQueryResolutionPolicy -Name "InsitutePrivate1ResolutionPolicy" -Action ALL
OW -ClientSubnet "eq,PrivateSubnet1" -ZoneScope "InstitutePrivateScope,1" -ZoneName "Dr
agon.Apollo"

Add-DnsServerQueryResolutionPolicy -Name "InsitutePrivate2ResolutionPolicy" -Action ALL
OW -ClientSubnet "eq,PrivateSubnet2" -ZoneScope "InstitutePrivateScope,1" -ZoneName "Dr
agon.Apollo"

Add-DnsServerQueryResolutionPolicy -Name "InstitutePublic1ResolutionPolicy" -Action ALL
OW -ClientSubnet "eq,PublicSubnet1 " -ZoneScope "InstitutePublicScope,1" -ZoneName Drag
on.Apollo

Add-DnsServerQueryResolutionPolicy -Name "InsitutePublic2ResolutionPolicy" -Action ALLO
W -ClientSubnet "eq,PublicSubnet2" -ZoneScope "InstitutePublicScope,1" -ZoneName "Drago
n.Apollo"

restart-service DNS -passthru
```

3.2.3. Name Resolution Tests

With the above policies in place, each of the subsequent trials attempted the following actions:

- Resolve the name cnn.com from RWDC, RODC, WEB using rwdc.dragon.apollo as the DNS server.
- Resolve the name www.dragon.apollo from RWDC, RODC, WEB using rwdc.dragon.apollo as the DNS server.
- Add the server “WEB” to the domain using rwdc.dragon.apollo as the DNS server.
- Add a new read-write domain controller to the domain using rwdc.dragon.apollo as the DNS server.
- Repeat the above tests using rodc.dragon.apollo as the DNS server.
- Use PowerShell to confirm the policies exist on rwdc.dragon.apollo.

- Use PowerShell to confirm the policies replicated to rodc.dragon.apollo.
- Explore file system, Active Directory using ADSIEdit, and Registry to find the location of the policies.

4. Findings and Discussion

The DNS zone scope policies function correctly; however, there are previously undocumented specific configurations and uses discovered.

4.1. DNS Server Recursion Scope Policy

Before implementing the recursion policy, web.dragon.apollo was tested to confirm cnn.com and google.com were accessible using the command line tool nslookup and using Internet Explorer to access the website directly. The DNS cache was then flushed; the policy implemented on rwdc.dragon.apollo, and the test repeated.

When the webserver IP address was 192.168.111.68, name resolution succeeded for cnn.com and google.com. When the webserver IP address was 192.168.111.168, the answer, "*** RWDC.dragon.apollo could not find cnn.com: Query refused" was received.

Switching the DNS server to rodc.dragon.apollo, the name resolution again succeeded when the webserver had either 192.168.111.68 and 192.168.111.168. This was an unexpected result.

There was no effect on the name resolution for all local A, CName, and SRV records for the dragon.apollo domain on either server. Both the .68 and .168 IP addresses returned all records queried on both rwdc.dragon.apollo and rodc.dragon.apollo.

4.2. Selective Query Answer Based on Source IP Address

Using rwdc.dragon.apollo as the DNS server and attempting name resolution from web.dragon.apollo using nslookup, the name www.dragon.apollo was queried. Prior to the configuration of the four DNS Server Query Resolution Policies for private subnets 1 and 2 and public subnets 1 and 2, the name was not found. Resolution failure was expected because the name web.dragon.apollo did not exist in DNS yet.

Configuring the polices and attempting name resolution from web.dragon.apollo gave expected results. IP addresses 192.168.111.20 and 192.168.111.68 produced the result of 192.168.111.68. IP addresses 192.168.111.168 and 192.168.111.200 produced the result of 192.168.111.168.

An additional test to confirm the policy efficacy was conducted by adding a standard A record for www.dragon.apollo with the IP address of 192.168.111.24. This address was not a response to any of the queries from any IP address if 192.168.111.69 was configured as the DNS server and web.dragon.apollo had an IP address in the 192.168.111.0/24 subnet. Removing the associated policy (InsitutePrivate2ResolutionPolicy) from address range 192.168.111.64/28 (PrivateScope2) and changing the IP address of web.dragon.apollo to 192.168.111.68 caused the answer to the query www.dragon.apollo to change to 192.168.111.24.

All of the above tests were conducted with the DNS server configured to 192.168.111.67 (rodc.dragon.apollo) on web.dragon.apollo. Tests included the replacement, removal, and replacement of the InsitutePrivate2ResolutionPolicy. Creating the standard A record for 192.168.111.24 caused IP address resolution for www.dragon.apollo to succeed. Then all queries from all IP addresses succeeded returning only the .24 address.

4.3. Service Record Queries

An attempt was made to add web.dragon.apollo to the Active Directory domain. The server was unable to resolve the query for a domain controller from within the domain join dialog box. Further testing using nslookup uncovered failures to resolve such names as _ldap._tcp.dragon.apollo, rwdc.dragon.apollo, and _ldap._tcp.pdc._msdcs.dragon.apollo. Switching the DNS server in nslookup to rodc.dragon.apollo allowed for the resolution of these names. A domain join was not performed in favor of additional testing.

The IP address of web.dragon.apollo was changed to each of the same addresses used in the Selective Query trials without success. After removal of the InsitutePrivate2ResolutionPolicy and changing the IP address of web.dragon.apollo to

192.168.111.68, which was controlled by the InsitutePrivate2ResolutionPolicy, it was then possible to resolve all the names above and add the server to the domain without issue.

4.4. Policy Replication

The previous testing leads to the theory that DNS policies were not replicated to the read-only domain controller. Get-DnsServerQueryResolutionPolicy was used to verify the following five policies existed on the read-write domain controller rwdc.dragon.apollo:

- InstitutePublic1ResolutionPolicy
- InsitutePublic2ResolutionPolicy
- InsitutePrivate1ResolutionPolicy
- InsitutePrivate2ResolutionPolicy
- SplitBrainRecursionPolicy

None of the policies were found on the read-only domain controller. The read-write domain controller was rebooted and confirmed fully restarted then the read-only domain controller was rebooted. Checking again for the policies, they were still absent.

A hypothesis was created that the read-only configuration was causing some issues replicating the policies. The server was demoted to a member server and re-promoted to a read-write domain controller. After the initial synchronization was confirmed complete, and standard DNS records were compared and confirmed using the DNS.msc management console, the same tests were conducted. The policies did not replicate.

4.5. Policy Location

These tests led to the question of where policies are stored. Checking the C:\Windows\System32\dns folder for files, no files of significance were found. This is not unusual because the zones are Active Directory integrated.

Next was an exploration of the DNS Application partitions using ADSIEdit.msc (IT Free Training, 2013). ADSIEdit.msc was connected to:

dc=ForestDNSZones,dc=dragon,dc=apollo

dc=DomainDNSZones,dc=dragon,dc=apollo

The usual DNS zone data was found, but no evidence of the scopes was present.

Searching the local registry on rwdc.dragon.apollo for the phrase "SplitBrainRecursionPolicy," the DNS Server key was discovered and contained all the policies and policy components. *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS Server*

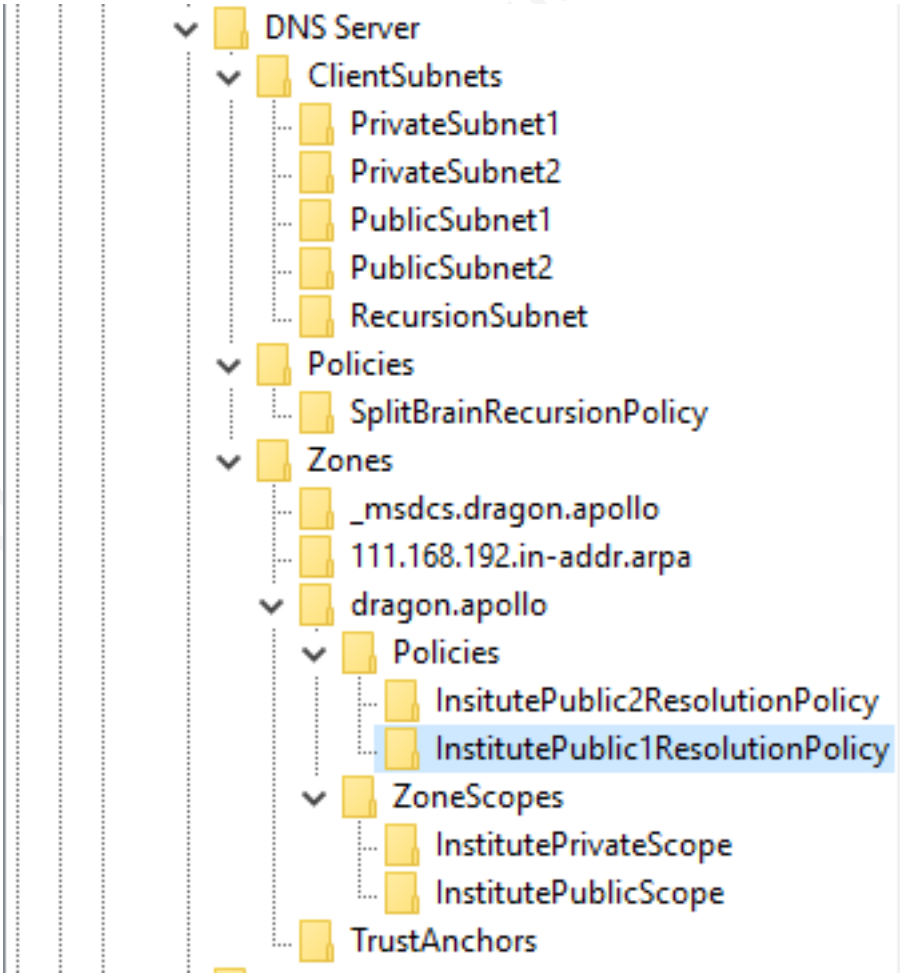

Figure 1 Registry tree of DNS Policies

Note that although one of the keys is named "Policies," these values have no corresponding Group Policies. Nor are there keys under *HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows* or *HKEY_LOCAL_MACHINE*

Mark J. Lucas, mark.lucas@student.sans.org

SOFTWARE\Policies\Microsoft\Windows NT that have corresponding DNS-related names.

The registry values can be retrieved by PowerShell as follows:

```
Get-DnsServerQueryResolutionPolicy -ZoneName dragon.apollo -Name "InstitutePublic1Resol
utionPolicy" | fl

Name            : InstitutePublic1ResolutionPolicy
Level           : Zone
AppliesOn       : QueryProcessing
Action          : Allow
Condition       : And
IsEnabled       : True
ProcessingOrder : 1
ZoneName        : dragon.apollo
Criteria        : DnsServerPolicyCriteria
Content         : DnsServerPolicyContent
```

Compared to the values in registry key *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\DNS Server\Zones\dragon.apollo\Policies\InsitutePublic2ResolutionPolicy* there is a one-to-one correspondence. There was no documentation found regarding these registry values; however, further research changing values in PowerShell would allow for the determination of REG-DWORD values corresponding to the human-readable values.

Name	Type	Data
(Default)	REG_SZ	(value not set)
Action	REG_DWORD	0x00000001 (1)
AppliesOn	REG_DWORD	0x00000000 (0)
ClientSubnet	REG_SZ	eq,PublicSubnet2
Condition	REG_DWORD	0x00000000 (0)
Content	REG_SZ	1,InstitutePublicScope
IsEnabled	REG_DWORD	0x00000001 (1)
ProcessingOrder	REG_DWORD	0x00000002 (2)

Figure 2 InstitutePublicResolutionPolicy Registry Entry

Mark J. Lucas, mark.lucas@student.sans.org

Literature searches were conducted for more information concerning these registry values, and no further information was found. Preliminary testing indicates these values can be directly created and changed in the registry. Care must be taken to exactly reference other keys. For example, the ClientSubnet value must exactly match the desired key under the ClientSubnet keys. For this reason, PowerShell is the preferred method of changing values. If an expected value is not found, the PowerShell command will fail and provide error feedback. There is no similar check in the Registry.

PowerShell does similar checks when removing policies or policy components. PowerShell, for example, does not permit the removal of a subnet definition used in an existing policy, whether or not the policy is enabled. Again, the Registry has no such check, and a policy component can be removed, which will render a given policy inoperative without warning.

5. Recommendations and Implications

Under particular circumstances, DNS Policies can be useful to limit DNS responses to queries and provide different answers to different source IP addresses. This can provide a high level of security. In particular, the Recursion Policy could prove very useful in an open environment. However, due to the current implementation of DNS Policies, the amount of labor required to configure the policies to be significantly effective in an open environment appears to be far too high to be practical.

Maintenance of the policies must be done by PowerShell, and dynamic DNS registration of any protected records must be disabled. Policies must be maintained individually on each separate DNS server.

Replacing public-facing WINS with an Active Directory integrated public-facing DNS server configured with restrictive policies is not viable.

Documentation of the exact configuration, as well as the PowerShell commands used, is recommended. The documentation should be a running list, including removals as well as adds.

Mark J. Lucas, mark.lucas@student.sans.org

5.1. Recommendations for Practice

5.1.1. Recursion

If DNS servers are exposed to the open internet, Recursion Policies are useful. It is possible to create policies to permit recursive lookups for organizational IP addresses and refuse recursion for all other IP addresses. The amount of maintenance on these policies is minimal, as organizational IP address space rarely changes.

Recursion policies are best utilized by turning off recursion for all sources and then selectively opening recursion as needed. Most examples found to turn on recursion based upon interface on the DNS server; however, using client subnet ranges works equally well. If there is more than one range needed, add ranges as needed, or create more than one recursion policy, using one policy per IP address range.

5.1.2. Selective Query Answer

Selectively answering DNS queries based on the source address is more problematic. If the server is stand-alone and internal vs. external addresses are required, and the DNS server holds a minimal number of records, or the records rarely change, this solution could be tenable. However, this research goal was to replace WINS with Active Directory-integrated DNS using Dynamic DNS for most registrations. The current state of DNS Policies creates a time-consuming manual maintenance situation.

The primary difficulty centers on the functionality that once a subnet range is defined and applied to a DNS Query Resolution Policy, all answers incoming from that subnet range can only be answered by values within that zone scope. No other values are provided for that subnet. In an Active Directory-integrated DNS environment, this leaves the administrator with two choices: 1) Define the entire outside world IP address space (IP v4 and v6) and apply the policies to those defined addresses, or 2) Define the internal IP address space and recreate all the A, CName, and SRV records. Option 2 removes all dynamic registration, which eliminates the ability to remove WINS unless every workstation has a static IP address - a rare and labor-intensive situation. Option 1 may be viable; however, further research and extensive penetration testing need to be performed before assessing the security and robustness of the solution. A downside to Option 1 is

that it would be impossible to selectively answer queries with private vs. public addresses if the DNS servers were located behind internal firewalls. Thus, this does not solve the public/private split-brain scenario.

5.1.3. Replication

Any decision regarding the use of DNS policies must include procedures to address the lack of replication. Because the PowerShell commands must run on each DNS server, procedures must be in place to record the commands and replicate them on each Active Directory DNS server. Each server must then be tested to ensure the provision of the correct responses to the appropriate client based on the source IP address. This lack of zone scope policy replication is a significant shortcoming in the viability of DNS Policies for extensive use. DNS is an indispensable component of all network connectivity and must, therefore, be redundant. Without policy replication, administrators' workload significantly increases, and the possibility of human error rises exponentially. If there is one reason to avoid the use of DNS Policies at this juncture, this is the shortcoming.

5.2. Implications for Future Research

The research topic that would provide the most immediate benefit is devising replication automation for the policies. This automation could take the form of a PowerShell script that ingested values, including server names and then creating the appropriate policies on all servers simultaneously. The script could also provide a log of actions, which would be the automatic documentation of changes. As an alternative, because the values are held in the Registry, a Group Policy module could be designed to apply the appropriate policies. Group Policy has the advantage of being automatically applied should a server be replaced or new server introduced. While this does not produce a log, it has the advantage of automatic application on all servers under the Group Policy. This greatly reduces the possibility of human error.

Investigation and penetration testing based on the idea of applying policies to all non-organizational IP addresses and allowing organizational IP addresses to access the DNS server, holds a great deal of merit.

Mark J. Lucas, mark.lucas@student.sans.org

A graphical interface that would permit the viewing, if not maintenance of the zone scopes, associated IP addresses, and records within each scope would be invaluable. This might be accomplished by a PowerShell script that could read the registry values.

Further inquiry into the split-brain scenario problem is worth addressing. What possible configurations could be used in this situation? How can different answers be applied without having to recreate the entire DNS infrastructure?

6. Conclusion

The ability to replace WINS with Dynamic DNS controlled by DNS Policy in an open environment is not currently viable. There are too many variables and shortcomings in the current functionality to make this a reality. However, DNS Policy is useful for stand-alone servers provided extensive documentation, change management procedures, and testing regimes are in place. Additional research and functionality improvements are required to make this tool viable for wide-spread implementation.

References

Center for Internet Security. (2019). *CIS Controls*. Retrieved from https://www.cisecurity.org/

Cisco Systems Inc. (2018). Configuring Inspection of Basic Internet Protocols. In *Cisco ASA 5500 Series Configuration Guide using the CLI, 8.4 and 8.6*. Retrieved from https://www.cisco.com/c/en/us/td/docs/security/asa/asa84/configuration/guide/asa_84_cli_config/inspect_basic.html#10154

Infoblox. (2018). DNS Security Best Practices. Retrieved July 4, 2020, from Infoblox DNS Security Resource Center website: https://www.infoblox.com/dns-security-resource-center/dns-security-best-practices/

IT Free Training. (2013). DNS and Active Directory Partitions - YouTube. Retrieved July 5, 2020, from https://www.youtube.com/watch?v=iKEbXBbG_VQ

Masterson, M., Knief, H., Vinick, S., & Roul, E. (1998). Integrating DNS with WINS. Retrieved June 27, 2020, from Microsoft Technet website: https://web.archive.org/web/20090403223319/http://technet.microsoft.com/en-us/library/cc722878.aspx

Microsoft Corporation. (1996). Microsoft Windows NT Server 4.0 WINS: Architecture and Capacity Planning. Retrieved June 28, 2020, from http://wwwdisc.chimica.unipd.it/luigino.feltre/pubblica/unix/winnt_doc/wins/msdn_winswp.html

Microsoft Corporation. (2020). DnsServer. Retrieved June 28, 2020, from Windows IT Pro Center website: https://docs.microsoft.com/en-us/powershell/module/dnsserver/?view=win10-ps

Mitchell, B. (2020). NetBIOS: What It Is and How It Works. Retrieved June 28, 2020, from Lifewire - Internet, Networking, & Security website: https://www.lifewire.com/netbios-software-protocol-818229

Mockapetris, P. (1987). Request for Comments: 1035 - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. *Network Working Group*,

November 1.

Pramatarov, M. (2018). DNS history. When and why was DNS created? - ClouDNS Blog. Retrieved June 28, 2020, from ClouDNS.net website: https://www.cloudns.net/blog/dns-history-creation-first/

Ross, E., & et al. (2020a). DNS Policy Scenario Guide | Microsoft Docs. Retrieved June 4, 2020, from Microsoft Documentation website: https://docs.microsoft.com/en-us/windows-server/networking/dns/deploy/dns-policy-scenario-guide

Ross, E., & et al. (2020b). IP Address Management (IPAM) | Microsoft Docs. Retrieved June 28, 2020, from Micro website: https://docs.microsoft.com/en-us/windows-server/networking/technologies/ipam/ipam-top

Ross, E., & et al. (2020c). Set Access Scope for a DNS Zone | Microsoft Docs. Retrieved June 13, 2020, from Microsoft Documentation website: https://docs.microsoft.com/en-us/windows-server/networking/technologies/ipam/set-access-scope-for-a-dns-zone

Ross, E., & et al. (2020d). Use DNS Policy for Split-Brain DNS in Active Directory | Microsoft Docs. Retrieved June 27, 2020, from Microsoft Network Documentation website: https://docs.microsoft.com/en-us/windows-server/networking/dns/deploy/dns-sb-with-ad

Ross, E., McIllece, J., Poggemeyer, L., & Yisheng, J. (2020). What's New in DNS Server in Windows Server | Microsoft Docs. Retrieved June 4, 2020, from Microsoft Documentation website: https://docs.microsoft.com/en-us/windows-server/networking/dns/what-s-new-in-dns-server

Last Updated: October 21st, 2020

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Sydney 2020	**Sydney, AU**	**Nov 02, 2020 - Nov 14, 2020**	**Live Event**
SANS Secure Thailand	**Bangkok, TH**	**Nov 09, 2020 - Nov 14, 2020**	**Live Event**
APAC ICS Summit & Training 2020	**Singapore, SG**	**Nov 13, 2020 - Nov 28, 2020**	**Live Event**
SANS Community CTF	**,**	**Nov 19, 2020 - Nov 20, 2020**	**Self Paced**
SANS Local: Oslo November 2020	**Oslo, NO**	**Nov 23, 2020 - Nov 28, 2020**	**Live Event**
SANS OnDemand	**OnlineUS**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**