

การเข้าใช้งานประยุกต์ Kerberos จาก โหมด IP เสมือน

Access Kerberized Application from Virtual IP Mode

พงษ์ศักดิ์ จงจิตต์

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต ปทุมธานี 12121

บทคัดย่อ

การนำ Kerberos เข้ามายึดในการ Authentication จะช่วยให้เครือข่ายมีความน่าเชื่อถือ มีความปลอดภัย สะดวกในการใช้ และสร้างความเป็นส่วนตัวให้แก่ข้อมูล แต่ Kerberos ไม่สนับสนุนความโปร่งใสของเครือข่าย โดยเฉพาะอย่างยิ่งการขอบริการข้ามเครือข่ายแบบ IP เสมือน(Virtual IP Mode) ดังนั้น เมื่อต้องการนำ Kerberos มาประยุกต์ใช้ จึงจำเป็นต้องปรับสภาพแวดล้อมให้เหมาะสม กับการทำงานของ Kerberos เก่านั้น ทำให้เกิดข้อจำกัดขึ้น

งานวิจัยนี้จึงต้องการนำวิธีการเพื่อทำให้ผู้ใช้สามารถขอบริการข้ามเครือข่ายแบบ IP เสมือนได้ โดยการระบุ IP ในฟอร์แมต ของ Credential

Abstract

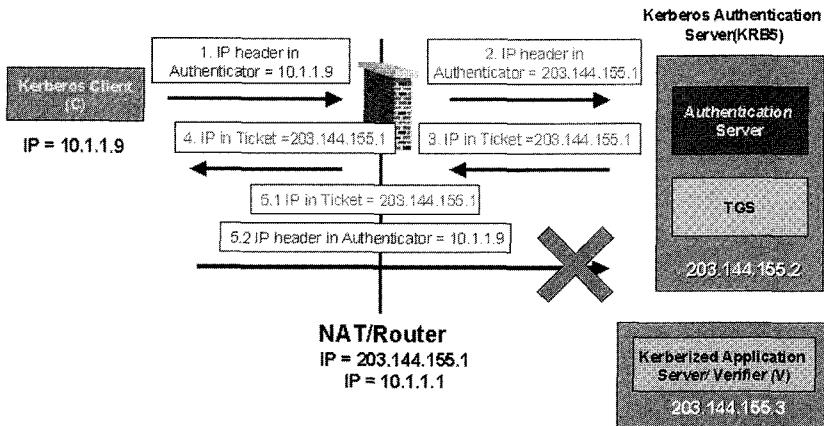
The use of Kerberos improves authentication, reliability, security, ease-of-use and data privacy. Although Kerberos has a number of benefits, it does not support network transparency. Especially, when a client requires a service via virtual IP mode. So, we need to configure the computer system to make it compatible with this limitation.

This research proposes a new way to overcome this limitation by identifying the NAT IP in the Kerberos credential.

1. บทนำ

Kerberos เป็นวิธีการ Authentication ที่เริ่มพัฒนาโดย MIT ซึ่งเป็นส่วนหนึ่งของโครงการ Athena และยอมรับว่า เป็นวิธีการ Authentication ที่มีความปลอดภัยสูง มีความน่าเชื่อถือ สามารถประยุกต์ใช้ได้กับหลากหลาย application โดย pragtic แล้วการที่ Kerberos จะทำงานได้ ต้องประกอบด้วยองค์ประกอบ 3 ส่วนด้วยกัน คือ Kerberos Client, Kerberos Authentication Server และ Kerberized Application Server [7-10] โดยจะเรียกแต่ละองค์ประกอบว่า Principle และเรียกขوبเขตที่ Kerberos นั้น ๆ ดูแลว่า Realm

Kerberos client เป็น principle ที่มีหน้าที่ในการขอ Authentication ไปยัง Kerberos Authentication Server ด้วยการส่ง Identity, Passphase และชื่อของ Verifier (หรือคือ Server ปลายทางที่ต้องการให้บริการ) และถ้าการ Authentication สำเร็จ จะได้รับ Session Key และ Ticket จาก Server กลับมา Kerberos Authentication Server เป็น principle ที่มีหน้าที่ในการควบคุมการ Authentication ทั้งหมด ซึ่งจะทำหน้าที่เป็น Key Distribution Center (KDC) เพื่อสร้าง Ticket ให้แก่ Client เมื่อ Kerberos Client ได้รับ Ticket จะนำ Ticket นี้ไปขอบริการจาก Verifier หรือคือ Kerberized Application Server นั้นเอง



ภาพที่ 1 ปัญหางานระหว่าง Kerberos และ NAT

จากภาพที่ 1 Service Ticket ที่ Authentication Server (AS) สร้างให้แก่ client จะถูก encrypt ด้วย Key ของ Kerberized Application Server เพราะฉะนั้น จึงมีเพียง Kerberized Application Server เท่านั้นที่สามารถ decrypt Key ใน Ticket ได้ และสามารถตรวจสอบว่า Ticket ที่ได้รับ ถูกต้องหรือไม่ นอกจากนี้ เพื่อป้องกันการ Replay Attack Kerberized Application Server ยังต้องตรวจสอบ Timestamp ใน Ticket ด้วยว่ายัง valid อยู่หรือไม่ ถ้า verify แล้วพบว่าเป็น Ticket ที่ถูกต้องและ Timestamp ใน Ticket ยังอยู่ในช่วงเวลาที่กำหนดจึงจะยอมให้บริการ

แต่การนำ Kerberos มาประยุกต์ใช้ยังมีข้อจำกัดบางประการ ยกตัวอย่างเช่น เมื่อผู้ใช้งานอยู่ใน Realm 1 ต้องการไปขอ Authentication จาก Authentication Server ที่อยู่ต่าง Realm (ในที่นี่สมมติให้เป็น Realm 2) C ส่ง Request เพื่อขอ Ticket (หรือ TGT) จาก TGS แต่เมื่อจาก Client C อยู่ด้านหลัง NAT ดังนั้น IP ของ Requestor จึงถูกเปลี่ยน (ปกติ NAT จะมี IP address 2 ชุด โดยจะ map IP address ระหว่างเครือข่าย 2 วง) เมื่อ Request message ส่งไปยัง TGS จะปรากฏ IP address ของ NAT เป็นผู้ขอ ดังนั้น TGS จึงสร้าง TGT โดยบรรจุ IP address ของ NAT แทนที่จะเป็น IP address ของ C

เนื่องจาก TGT ที่ได้จะถูก encrypt โดย Secret Key จาก Kerberized Application Server ดังนั้น เมื่อ C ได้รับ TGT จึงไม่สามารถ decrypt และตรวจสอบข้อมูลใน TGT ได้

จากนั้น C จะส่ง Authenticator ที่บรรจุ IP address ของตัวเองไปขอサービスจาก Kerberized Application Server โดยส่ง TGT (ที่มี IP address ของ NAT) ไปด้วย เมื่อ Credential (Authenticator + TGT) ถูกส่งไปยังปลายทาง (Kerberized Application Server) Server ปลายทางจะ decrypt คำใน Credential และเปรียบเทียบกัน แต่เนื่องจาก TGT บรรจุ IP address ของ NAT แต่ Authenticator บรรจุ IP address ของ C ทำให้คำ Checksum ที่ได้ไม่ถูกต้อง Kerberized Application Server จึงมองว่า Credential ที่ส่งมาไม่ถูกต้อง ทำให้ drop Request message นั้นไปในที่สุด

จากปัญหาดังกล่าว จึงเป็นสาเหตุที่ทำให้ผู้ใช้ไม่ได้รับความปลอดภัยซึ่งเป็นคุณสมบัติที่สำคัญใน Distributed system ดังนั้น ถ้าเราสามารถหาแนวทางเพื่อทำให้ Kerberos สนับสนุน การทำงานดังกล่าวได้ ก็จะเป็นการเพิ่ม Transparency ให้แก่ Kerberos เมื่อจากในปัจจุบัน เครือข่ายขนาดใหญ่ลักษณะแบบเน็ตเวิร์ก มีการแบ่งเน็ตเวิร์กออกเป็นส่วน ๆ (Network Segmentation) และติดตั้ง Firewall/NAT เพื่อควบคุมและป้องกันการเข้า-ออกของ packet โดยติดตั้ง Kerberos Authentication Server และ Kerberized

Application server กระจายอยู่ในแต่ละเครือข่าย ผู้ใช้จากเครือข่ายหนึ่งสามารถขอ Authentication ไปยัง Server ต่างๆ เครือข่ายที่อาจอยู่ห่างไกลออกไป โดยมีต้องคำนึงถึง NAT กันระหว่างกลางหรือไม่ก็จะทำให้เครือข่ายมีความไม่ร่วงโรยมากยิ่งขึ้น

2. งานวิจัยที่เกี่ยวข้อง

แนวความคิดแรก เป็นแนวความคิดที่มุ่งเน้นการแก้ปัญหา Virtual Private Network เพื่อให้สนับสนุนการทำงานของ NAT พัฒนาโดยทีมงานจาก IBM เมื่อปี 2002 [13] เนื่องจากปรกติแล้ว เมนู NAT จะช่วยป้องกันการ access จากเครือข่ายภายนอกโดยการซ่อน IP address ภายใต้เครือข่ายไว้ แต่วิธีการดังกล่าวก็อาจทำให้เราไม่สะดวกในการใช้ application บางอย่าง อาทิ IPsec ซึ่งเป็นหัวใจหลักในการทำ Tunnel ให้แก่ VPN เมื่อจากการทำงานของ IPsec จำเป็นต้องระบุ IP address ต้นทางและปลายทางก่อนสร้าง Tunnel ทุกครั้ง เมื่อมี NAT ทั้งกลาง NAT จะแปลง IP address ของต้นทางให้เปลี่ยนไปจากเดิม ทำให้ปลายทางจะเห็น IP address ที่แตกต่างกันที่ระบุไว้ ดังนั้น ปลายทางจะ drop packet ตั้งกล่าวไว้ และทำให้เราไม่สามารถทำ Tunnel ได้ ซึ่งส่งผลให้เราไม่สามารถทำ Virtual Private Network ได้เช่นกัน เพื่อให้เราสามารถทำ VPN ขั้ม NAT ได้ IBM จึงเสนอวิธีการสร้าง header ใหม่ เรียกว่า IP/UDP header ซึ่งมีหน้าที่ encapsulate IPsec ให้ไปใน User Datagram Protocol, UDP โดย header ใหม่นี้จะสามารถทะลุผ่าน NAT ไปยังปลายทางได้โดยไม่จำเป็นต้องแก้ไข Policy ใด ๆ ของ NAT เลย เวิธีการนี้ว่า UDP Encapsulation เมื่อ packet ลังกลากลูกสั่นไปยังปลายทางแล้ว header จะถูก decapsulate ออกมานะ ซึ่งจะได้ IPsec ตามเดิม และสามารถทำงานได้ตามปกติ เมื่อ IPsec สามารถทะลุผ่าน NAT ไปได้ ก็เท่ากับเราสามารถทำ tunnel ให้แก่ Virtual Private Network ได้

แต่วิธีการดังกล่าวจะเห็นว่ามีความซับซ้อนและยุ่งยากมาก เนื่องจากเรามาจำเป็นต้องสร้าง Protocol header ใหม่ขึ้นมา และวิธีการตั้งกล่าวถือว่าบังเอิญเป็น proprietary และ VPN ของ IBM วิ่งด้วย ทำให้การใช้งานนั้นจำกัดแค่ผลิตภัณฑ์บางกลุ่มเท่านั้น ยังไม่ได้รับการสนับสนุนให้เป็นมาตรฐานเท่าที่ควร

MIT ซึ่งเป็นต้นกำเนิดของ Kerberos เองก็เล็งเห็นและหาแนวทางในการแก้ปัญหาระหว่าง Kerberos และ NAT ชนิด กัน โดยเสนอแนวทางที่จะทำให้ Ticket ที่ใช้สื่อสารระหว่าง Client และ Kerberos Authentication Server มีส่วนให้ IP address [20] โดยกำหนดให้การระบุ IP address ใน Ticket เป็น Option ดังนั้น ก่อนที่เราจะขอ Authentication ไปยัง Server เราสามารถ disable IP address ออกไป เมื่อเราไม่มี IP address ดังนั้น เมื่อ Authentication Request ส่งไปยัง Kerberos Authentication Server โดยไม่ระบุ IP address ต้นทาง Kerberos Authentication Server ก็จะสร้าง Service Ticket ที่ไม่มี IP address กลับไป วิธีการ ดังกล่าว ปัจจุบัน ถูกพัฒนาเป็น feature หนึ่งใน Kerberos Ticket Manager และทำให้การใช้งานสะดวกยิ่งขึ้น เมื่อจาก เรายังจำเป็นต้อง กันเวลาเรื่อง IP address อีกด้วย โดยการสื่อสารระหว่าง Client, NAT, Authentication Server และ Kerberized Application จะอาศัย Identity อื่นใน Ticket แทน เมื่อ Kerberized Application Server ได้รับ Credential ซึ่งประกอบด้วย Authenticator และ Ticket (TGT) ที่ไม่ระบุ IP address และผ่านการตรวจสอบ Identity ส่วนอื่นใน Credential และ Kerberized Application Server ก็สามารถให้บริการได้ทันที

แต่เมื่อจากขั้นตอนการใช้งานดังกล่าวไม่มีการตรวจสอบ IP address เลย ดังนั้น ในอนาคตถ้าวิธีการดังกล่าวได้รับความนิยมเพิ่มมากขึ้น อาจมีผู้คิดค้นเครื่องมือในการตักกับ TGT และนำมายังเครื่องมือในการจำลองตนของเรา User เพื่อขอให้บริการจาก Remote Kerberized Application Server เพราฉะนั้น วิธีดังกล่าวอาจกลายเป็นจุดอ่อนของ Kerberos ในอนาคตได้ นอกจากนี้ Kerberos Authentication Server และ Kerberized Application Server ที่ Remote site ยังไม่สามารถ Tracking IP address ของ User ขาด บริการจากที่ได้ ทำให้ยากต่อการตรวจสอบ และ/หรือ บริการงานในอนาคต

อีกแนวความคิดหนึ่ง คือการทำให้ Client สามารถ access ไปยัง Service ยัง Kerberos Authentication Server และ Kerberized Application Server ที่อยู่ต่าง Realm ได้ โดยวิธีการดังกล่าวมีการคิดค้นและพัฒนาอย่างต่อ

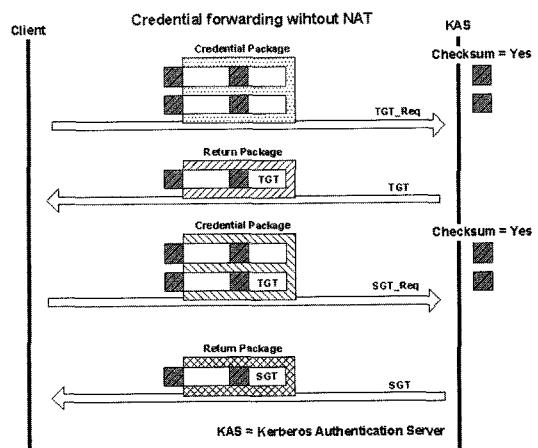
เพื่องและมีประโยชน์อย่างมาก เพราะ Client สามารถขอรับการได้ทั้ง Realm ที่อยู่ใกล้เคียงหรือแม้แต่ Realm ที่อยู่ห่างไกล ออกไปร่วมกับล่า เรียกว่า Cross-realm Authentication [11]

Cross-realm Authentication เป็น Feature หนึ่งซึ่งถูกพัฒนาขึ้นตั้งแต่ Kerberos-4 และได้ปรับเปลี่ยนรูปแบบไปใน Kerberos-5 เพื่อให้การทำงานมีประสิทธิภาพมากขึ้น Cross-realm จะยอมให้ Ticket ที่ได้จาก Authentication Server หรือ TGT สามารถ forward ไปยัง Realm อื่น ได้ โดยผ่าน Ticket Granting Server (TGS) ใน Realm นั้น ๆ แต่มีข้อกำหนดที่ว่า แต่ละ Realm จะเป็นต้องสร้างความสัมพันธ์ที่เชื่อมต่อได้หรือที่เรียกว่า Transitive Trust Relationship ก่อน

แต่เมื่อเรานำ Cross-realm authentication ไปใช้งาน เราต้องบวกกลับภัยด้านประการ ซึ่งอาจถือได้วาเป็นทั้งข้อดีและข้อเสียในเวลาเดียวกัน กล่าวคือ ถ้าเครือข่ายของเราต้องการ enable Cross-realm domain เราพบว่า sub-domain ได้ก็ตามที่อยู่ภายใต้ domain นั้นจะมีความเรื่องอิหร่วงกันโดยอัตโนมัติ หรือจะมี Transitive Trust Relationship ระหว่างกันทันที ตัวอย่างเช่น เมื่อเรา enable domain TU.AC.TH จะทำให้ sub-domain A.TU.AC.TH, B.TU.AC.TH และ C.TU.AC.TH มี Transitive Trust relationship ระหว่างกัน โดยอัตโนมัติ ซึ่งในบางกรณี เราอาจไม่ต้องการให้ Server A และ B Trust กัน เมื่อจากเหตุผลบางประการ ดังนั้น แม้ว่า Cross-realm domain จะมีประสิทธิภาพในการสื่อสารข้าม Realm กันได้ ทำให้เราสามารถขอรับบริการจาก Server ต่าง Realm แต่ถ้าระบบใด หรือหน่วยงานใดไม่ต้องการให้ sub-domain นั้นมี Trust Relationship กันเอง ก็อาจกล่าวเป็นข้อเสียได้เช่นกัน

นอกจากนี้ ถ้าเรานำ Cross-Realm Authentication ไปประยุกต์ใช้ร่วมกับการ Authentication แบบที่ไม่มี IP address ดังที่กล่าวไปแล้วข้างต้น ก็จะทำให้ User สามารถขอรับบริการไปยัง Kerberized Application Server ที่ไม่ออก IP โดยไม่ระบุ IP address ดังนั้น จะทำให้เราไม่สามารถตรวจสอบได้ว่า User ที่ขอรับบริการมาจาก Local หรือ Remote network ซึ่งวิธีดังกล่าว อาจเป็นการเพิ่มปัญหาในด้านความปลอดภัยและไม่สามารถบริหารทรัพยากรได้

จากตัวอย่างที่ได้กล่าวไปแล้วข้างต้นคือ ตัวอย่างของปัญหาและแนวคิดในการแก้ปัญหาซึ่งอาจเกี่ยวข้องกับงานวิจัย โดยตรง หรืออาจเป็นเพียงแนวความคิดที่ผู้วิจัยใช้เป็นฐานความรู้ในการศึกษาและพัฒนาเพื่อเป็นแนวทางในการวิจัยต่อไป ซึ่งจะเห็นว่ากิจกรรมบางอย่าง แม้ว่าจะสามารถแก้ปัญหาเกี่ยวกับ Kerberos และ NAT ได้ แต่เราอาจต้องยอมรับในจุดอ่อนหักอื่น ที่ตามมาและอาจกล่าวเป็นจุดอ่อนของ Kerberos ในอนาคต



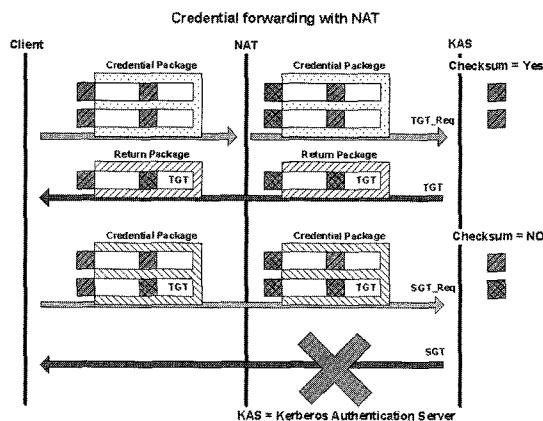
ภาพที่ 2 แสดง Credential ในกรณีไม่มี NAT

3. วิธีการดำเนินงานวิจัย

3.1 แนวความคิดที่ใช้ในการวิจัย

จากปัญหาที่ได้กล่าวไปแล้วในบทที่ 1 เมื่อเรามีการอนุญาตให้เครือข่ายที่อยู่ภายใต้ NAT ที่มีหน้าที่หลักในการแปลง IP header ที่อยู่ภายใต้เครือข่าย ให้กลับเป็น IP address ที่อยู่ทาง โดยมองเป็นปะโยชน์ในการซ่อน IP address เป็นหลัก ดังนั้น ถ้าเราสามารถทำให้ Kerberos Authentication Server และ Kerberized Application Server ที่อยู่ต่างเครือข่ายรู้จัก IP address ของ NAT และใช้ IP address ดังกล่าวเป็นส่วนหนึ่งในการตรวจสอบ Credential ก็จะทำให้ Kerberos Authentication Server และ

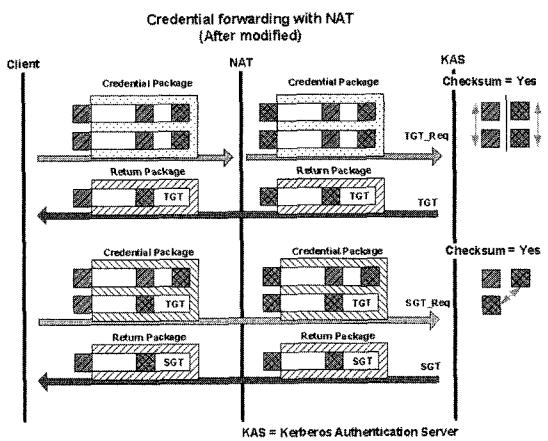
Kerberized Application ยอมให้บริการตามที่ Client ร้องขอ และยังคงความปลอดภัยในการ Authentication เกมีข้อดีมี



ภาพที่ 3 แสดง Credential ในกรณี NAT

ภาพที่ 3 แสดงให้เห็นว่า เมื่อ Credential ซึ่งประกอบไปด้วย Authenticator และ TGT ถูกส่งไปยัง Keberos Authentication Server และ Kerberized Application จะ decrypt Credential พร้อมกับตรวจสอบ IP address ที่อยู่ภายใน Authenticator และ TGT ซึ่งในที่นั้นพบว่า IP address ของ Authenticator และ TGT ไม่ตรงกัน

ในที่นี้ ผู้จัดจะแก้ปัญหาดังกล่าว โดยเพิ่ม IP address ของ NAT ใน Credential เมื่อ Client ส่ง TGT_Request ซึ่งบรรจุ IP address ของ Client และ NAT ไปยัง Kerberos Authentication Server KDC จะสร้าง TGT ที่บรรจุ IP address ของ Client และ NAT ลงไว้ด้วย

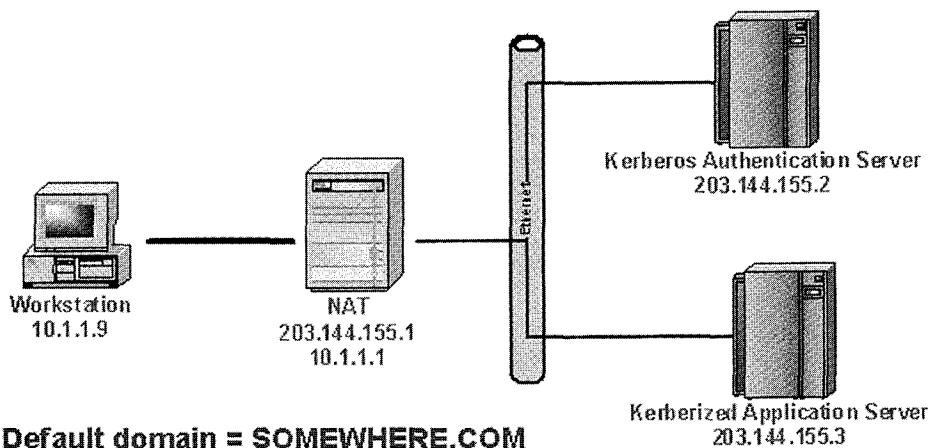


ภาพที่ 4 แสดงแนวทางการแก้ปัญหาโดยเพิ่ม IP address

เมื่อ Client ต้องการขอริการเบอร์ยัง Kerberized Application Server Client จะส่ง Credential ซึ่งประกอบด้วย Authenticator และ TGT ไปยัง TGS (ใน Kerberos Authentication Server) และจะได้รับ SGT ที่บรรจุ IP address ของ Client และ NAT กลับมาพร้อมกับส่ง SGT ดังกล่าวไปยังปลายทาง เมื่อ Kerberized Application Server ปลายทางตรวจสอบ จะยอมให้ Compromise หรือยอมให้บริการได้ หากพบว่า IP address ใน Credential ตัวใดตัวหนึ่ง ตรงกัน โดย NAT จะมีหน้าที่ในการ Mapping ระหว่าง IP ภายในของ Client (IP ปลอม) กับค่า Binary ของ Ticket แต่ละตัว (Ticket มีการส่งในรูปของ Binary อยู่แล้ว) ทั้งนี้เพื่อป้องกันไม่ให้มีการ Replay attack ในกรณีที่ลักลอบนำ TGT ดังกล่าวไปใช้

3.2 การจำลองโครงงาน

ผู้วิจัยจะจำลองการทำงานตามมาตรฐานความคิดดังกล่าว ดังภาพที่ 5 ซึ่งจะเห็นว่า NAT กับกลางระหว่าง Client C



ภาพที่ 5 Project Diagram

ชื่อยูนิเครือข่าย 10.1.1.x และต้องการขอรับการไปยัง

Kerberized Application Server ที่อยู่ในเครือข่าย
203.144.155.x ซึ่งมี Kerberos Authentication Server ทำ
หน้าที่เป็น TGS
ก้าหนดให้

IP address ของ C = 10.1.1.9

IP address ของ NAT = 10.1.1.1, 203.144.155.1

IP address ของ Kerberos Authentication Server

= 203.144.155.2

IP address ของ Kerberos Application Server

= 203.144.155.3

และสร้าง Realm ชื่อ SOMEWHERE.COM เพื่อใช้
เป็น default domain และเพิ่ม field extra_address โดย
บรรจุ IP address ของ NAT ใน file

[root@workstation /]# /etc krb5.conf

โดยมีรายละเอียด ดังนี้

[libdefaults]

proxiable = true

forwardable = true

default_realm = SOMEWHERE.COM

noaddresses = false

extra_address = 203.144.155.1

[realms]

SOMEWHERE.COM = {

kdc = 203.144.155.2:88

default_domain = SOMEWHERE.COM}

[domain_realm]

.somewhere.com = SOMEWHERE.COM

somewhere.com = SOMEWHERE.COM

[logging]

kdc = CONSOLE

จากนั้น ได้พัฒนาโปรแกรมเพื่อทำให้ Credential
สนับสนุน field extra_address โดยในที่นี่ ผู้ใช้ได้ปรับปรุง
เฉพาะ Credential ของ Kerberos 5 เท่านั้น (ปกติแล้ว
Credential สามารถสนับสนุนทั้ง Kerberos 4 และ 5) โดยแก้
ไข file ดังกล่าวใน

[root@workstation /]# /kinit.c

เมื่อเริ่มการใช้งาน Client จะส่ง AS_Request ไปยัง
Kerberos Authentication Server ซึ่งจะได้รับ TGT ที่บรรจุ
IP address ของทั้ง Client และ NAT และใช้ TGT ดังกล่าว
ในการขอ Service Ticket (หรือที่เรียกว่า SGT) เพื่อสามารถเข้าใช้
บริการยัง Kerberized Application Server

ในการทดสอบ ถ้าการทำงานทุกขั้นตอนถูกต้อง เมื่อ Client ผ่านการ Authentication ในครั้งแรกแล้วได้รับ Ticket-Granting Ticket(TGT) และ Client สามารถใช้บริการจาก Kerberized Application ได้เลย โดย Client จะจำเป็นต้องผ่านกระบวนการ Authentication อีกชั้นเป็นผลมาจากการ Client ได้ส่ง TGT ที่ตนได้รับไปยัง TGS เพื่อขอ Service Ticket ไปยัง Kerberized Application Server จากการทำงานดังกล่าว จะเห็นว่า Client จะได้รับความสะดวกเมื่อจาก Client ไม่จำเป็นต้องจดจำ Password ไม่แต่ล่ะ Application และกล่าวได้ว่า Kerberos สนับสนุนการทำงานแบบ OTP

ดังนั้น เมื่อ Client สามารถใช้งาน Kerberized Application เร毅力บว่า Client ได้รับ Ticket 2 ตัวด้วยกัน Ticket ตัวแรก ได้แก่ TGT ที่ได้จาก AS_Request และ Ticket ตัวที่สองคือ SGT ที่ใช้ขอบริการไปยัง Kerberized Application นั้นเอง (กำหนดให้ app.somewhere.com เป็น Kerberized Application domain)

จากภาพที่ 6 จะเห็นว่าภายใน Ticket ประกอบไปด้วย IP address ของทั้ง Client และ NAT ซึ่งจะทำให้ Kerberized Application Server ยอมให้ Credential ดังกล่าวผ่านการตรวจสอบ และให้บริการในที่สุด

4. ผลของการวิจัย

4.1 ปัญหาระหว่าง Kerberos และ NAT

ในกรณีที่ Client ภายใต้ NAT ต้องการขอรับบริการไปยัง Kerberized Application ภายนอก NAT เมื่อ Client จะได้รับ Ticket หลังจากการ Authentication แต่ก็ไม่สามารถนำ Ticket นั้นไปใช้ได้ เนื่องจาก IP address ใน Credential ไม่ตรงกัน โดยผลลัพธ์ที่ได้ แสดงดังภาพที่ 7

โดยเราสามารถพิจารณาผลลัพธ์ที่ได้จาก Log file ใน Kerberos Authentication Server ได้ที่

```
[root@firewall ~]#tail -f /var/log/krb5kdc.log
```

ดังแสดงในภาพที่ 7 แสดงให้เห็นว่า เมื่อเราได้รับ TGT (แสดง IP 10.1.1.9) จากการ Authentication แต่เมื่อขอ บริการไปยัง Kerberized Application Server กลับพบว่าไม่มี IP address (แสดง ERROR ว่า Incorrect net address) ซึ่งเป็นผลมาจากการ NAT เปลี่ยน IP address ไปแล้ว ตั้งแต่กระบวนการขอ Authentication (ใน Ticket บรรจุ IP address ของ NAT ไม่ใช่ IP address ของ Client)

```
Linux RedHat - [Ctrl+Alt+F1] - VMware Workstation
File Power Settings Devices View Help
Power Off Power On Suspend Reset Full Screen
Connection closed by foreign host.
[root@workstation root]# klist -a
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: tonkrau@SOMEWHERE.COM

Valid starting     Expires          Service principal
05/02/04 19:37:33  05/03/04 19:33:52  krtgt/SOMEWHERE.COM@SOMEWHERE.COM
                  Addresses: 203.144.155.1, 10.1.1.9
05/02/04 19:40:54  05/03/04 19:33:52  host/app.somewhere.com@SOMEWHERE.COM
                  Addresses: 203.144.155.1, 10.1.1.9

Kerberos 4 ticket cache: /tmp/ktkt0
klist: You have no tickets cached
[root@workstation root]#
```

ภาพที่ 6 แสดง Ticket ทั้งหมดที่ Client ได้รับเมื่อเข้าใช้ Kerberized Application สำเร็จ

```

@Linux_RedHat9-[Cn-AK-F1]-VMware Workstation
File Power Settings Devices View Help
Power Off Power On Suspend Reset Full Screen
root@workstation root# klist -a
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: tonkra@SOMEWHERE.COM

Valid starting Expires Service principal
05/02/04 19:44:13 05/03/04 04:42:45 krbtgt/SOMEWHERE.COM@SOMEWHERE.COM
    Addresses: 18.1.1.9

Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
root@workstation root# telnet -a -x -f -l tonkra app.somewhere.com
Trying 203.144.155.3...
Connected to app.somewhere.com (203.144.155.3).
Escape character is '^'.
Waiting for encryption to be negotiated...

Negotiation of authentication, which is required for encryption,
has failed. Good-bye.
[root@workstation root]#

```

ภาพที่ 7 ปัญหาระหว่าง Kerberos และ NAT

```

@Linux_RedHat9-[Cn-AK-F1]-VMware Workstation
File Power Settings Devices View Help
Power Off Power On Suspend Reset Full Screen
root@workstation root# 
root@workstation root# 

May 02 19:49:58 firewall krb5kdc[1309](info): AS_REQ (7 etypes {18 17 16 23 1 3
2}) 203.144.155.1: ISSUE: authtime 1083502198, etypes {rep=16 tkt=16 ses=16}, to
tonkra@SOMEWHERE.COM for krbtgt/SOMEWHERE.COM@SOMEWHERE.COM

May 02 19:50:11 firewall krb5kdc[1309](info): TGS_REQ (1 etypes {1}) 203.144.155
.1: PROCESS_TGS: authtime 0, <unknown client> for host/app.somewhere.com@SOMEWH
ERE.COM, Incorrect net address
May 02 19:50:11 firewall krb5kdc[1309](info): TGS_REQ (1 etypes {1}) 203.144.155
.1: PROCESS_TGS: authtime 0, <unknown client> for host/app.somewhere.com@SOMEWH
ERE.COM, Incorrect net address
May 02 19:50:11 firewall krb5kdc[1309](info): TGS_REQ (1 etypes {1}) 203.144.155
.1: PROCESS_TGS: authtime 0, <unknown client> for host/app.somewhere.com@SOMEWH
ERE.COM, Incorrect net address

```

ภาพที่ 8 แสดง ERROR ใน Log File

4.2 วิธีการแก้ปัญหาระหว่าง Kerberos และ NAT โดยไม่ใช้ IP address

จากปัญหาดังกล่าว MIT ได้เสนอวิธีการแก้ปัญหาโดยIgnore IP address ในทุกรอบวนการ ท้าให้รานั่งจำเป็นต้องใช้ IP address ที่ปั้นใน Credential และการตรวจสอบ ดังภาพที่ 9

จะแสดงให้เห็นถึง Ticket ที่ได้รับว่าไม่แสดง IP address เลย (แสดงเป็น none) และเมื่อขอบริการเปลี่ยน Kerberized Application Server ปลายทาง ก็สามารถใช้บริการได้ตามปกติ

```

@Linux_RedHat9 [Ctrl-Alt-F1] - VMware Workstation
File Power Settings Devices View Help
Power Off Power On Suspend Reset Full Screen
[roo...@workstation etc]# kinit -5 tonkra
Password for tonkra@SOMEWHERE.COM:
[roo...@workstation etc]# klist -a
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: tonkra@SOMEWHERE.COM

Valid starting     Expires          Service principal
05/01/04 01:48:47  05/01/04 18:47:08  krbtgt/SOMEWHERE.COM@SOMEWHERE.COM
Addresses: (none)

Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
[roo...@workstation etc]# telnet -a -f -x -l tonkra app.somewhere.com
Trying 18.1.1.5...
Connected to app.somewhere.com (18.1.1.5).
Escape character is '^]'.
Waiting for encryption to be negotiated...
[ Kerberos V5 accepts you as ``tonkra@SOMEWHERE.COM'' ]
[ Kerberos V5 accepted forwarded credentials ]
done.
Last login: Sat May  1 08:04:05 from 18.1.1.254
[tonkra@kerberize tonkra]$ 

```

ภาพที่ 9 การใช้บริการ Kerberized Application โดยไม่ใช้ IP address

```

@Linux_RedHat9 [Ctrl-Alt-F1] - VMware Workstation
File Power Settings Devices View Help
Power Off Power On Suspend Reset Full Screen
[roo...@workstation etc]# kinit -5 tonkra
Password for tonkra@SOMEWHERE.COM:
[roo...@workstation etc]# klist -a
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: tonkra@SOMEWHERE.COM

Valid starting     Expires          Service principal
05/02/04 20:00:36  05/03/04 04:58:24  krbtgt/SOMEWHERE.COM@SOMEWHERE.COM
Addresses: 203.144.155.1, 18.1.1.9

Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
[roo...@workstation etc]# telnet -a -f -l tonkra app.somewhere.com
Trying 203.144.155.3...
Connected to app.somewhere.com (203.144.155.3).
Escape character is '^]'.
Waiting for encryption to be negotiated...
[ Kerberos V5 accepts you as ``tonkra@SOMEWHERE.COM'' ]
[ Kerberos V5 accepted forwarded credentials ]
done.
Last login: Sun May  2 19:59:41 from 203.144.155.1
[tonkra@kerberize tonkra]$ 

```

ภาพที่ 10 แสดงผลลัพธ์ที่ได้ ภายหลังการเพิ่ม IP address ใน credential

โดย Log File จะมีการแสดง ERROR ได ๆ เช่นกัน เมื่อเวลาทำการรับ IP ของ NAT ใน Ticket เนื่องจากวิธีการตั้งค่าไม่เน้น IP address ไปใช้ในการตรวจสอบ

4.3 การแก้ไขภาระระหว่าง Kerberos และ NAT โดยใช้ IP address

ให้วิธีการก่อนหน้านี้ เมื่อเวลาสามารถแก้ปัญหาระหว่าง Kerberos และ NAT ได้ แต่วิธีการดังกล่าว อาจไม่เหมาะสมกับบาง Application ที่จำเป็นต้องระบุ IP address อีก ที่ VPN เป็นต้น ซึ่งจากการแก้ปัญหานี้ให้วิธีที่สองอาจเป็นสาเหตุหนึ่งที่ทำให้การพัฒนา Kerberized Application ยังอยู่ในวงที่จำกัด

ดังนั้น ผู้จัดจึงเสนอวิธีที่นักวิชาสามารถแก้ปัญหาระหว่าง Kerberos และ NAT ได้แล้ว ยังสามารถระบุ IP address ของทั้ง Client และ NAT ลงไว้ใน Credential ได้ โดยวิธีการนี้ อาจเป็นแนวทางในการพัฒนา Kerberized Application ที่จำเป็นต้องใช้ IP address ประกอบการใช้งานต่อไป

เมื่อพิจารณาจาก Log File พบว่า เมื่อมี IP address ใน Ticket 2 IP แต่ใน Log File ยังคงแสดง IP address เพียง IP ของ NAT เท่านั้น และ Client ยังสามารถขอรับการไปยัง Kerberized Application Server ได้ตามปกติ

5. สรุปผลงานวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการศึกษาปัญหาระหว่าง Kerberos และ NAT แสดงให้เห็นว่า แต่ละวิธีต่างก็มีข้อดีและข้อเสียในด้านความปลอดภัยที่แตกต่างกัน ขึ้นกับผู้พัฒนาว่าจะนำวิธีการใดมาใช้เพื่อให้เหมาะสมกับสภาพแวดล้อมของตนเอง

หากเราต้องการใช้ NAT เพื่อควบคุมความปลอดภัยจากรายงานออกเครือข่าย โดยป้องกันไม่ให้บุคคลภายนอกสามารถ access ไปยัง Client ปลายทางโดยตรง เราจะได้ประโยชน์จากการควบคุมความปลอดภัยด้าน Network Filtering เนื่องจากเราสามารถควบคุมและป้องกัน Port และ IP address ที่เหลือผ่านเครือข่ายได้อย่างมีประสิทธิภาพ

หรือถ้าเราต้องการประโยชน์ที่ใช้ Kerberos ซึ่งเป็นวิธีการควบคุมความปลอดภัยด้านการ Authentication เราเก็บไว้ใน "ผู้ที่ขอใช้บริการเป็นผู้ที่มี Authorization" จริง มีความปลอดภัยเพียงพอและสะดวกแก่ผู้ใช้ เมื่อจาก Kerberos สนับสนุนการ Authentication แบบ OTP ซึ่งถูกในระบบ (หรือแต่ละ Realm) นั้น ๆ มี Kerberized Application มา ก็จะยิ่งเพิ่มความสะดวกแก่ผู้ใช้ เนื่องจากผู้ใช้ไม่จำเป็นต้องจำ Username และ Password มา กและยังรองรับการขยาย (Scalability) ของระบบในอนาคตได้ โดยอาจเพิ่ม Kerberos Authentication Server ได้มากกว่าหนึ่งตัว หรือเพิ่ม Kerberized Application Server ให้อีกในอนาคต

แต่เมื่อเรานำระบบความปลอดภัยทั้งสองวิธีนี้มาประยุกต์ร่วมกันกลับกลายเป็นปัญหา เนื่องจาก Kerberos

พัฒนาขึ้นเพื่อเน้นการใช้งานภายในเครือข่ายเดียวกันเป็นหลัก ทำให้ไม่สนับสนุนการใช้งานกรณีต้นทางและปลายอยู่ต่างเครือข่ายกัน หรือถ้าเราต้องการทั้งสองวิธีมาประยุกต์ร่วมกันก็จำเป็นต้อง Compromise ความปลอดภัยลง โดยใน IP address มาใช้ประกอบการตรวจสอบ ซึ่งวิธีการดังกล่าว อาจไม่เหมาะสมกับ Application บางชนิดที่จำเป็นต้องระบุ IP address เป็นต้น

ดังนั้น ผู้จัดจึงนำเสนอและพัฒนาวิธีการทำงานร่วมกันระหว่าง Kerberos และ NAT โดยยังคงใช้ค่า IP address ใน การตรวจสอบ เพื่อใช้เป็นอีกแนวทางในการพัฒนา Kerberized Application ที่จำเป็นต้องระบุ IP address อีก อาทิ VPN เป็นต้น วิธีการดังกล่าว ถือได้ว่าเป็นการเพิ่มประสิทธิภาพให้แก่ Kerberos และรองรับการพัฒนา Kerberized Application ที่ หลากหลายมากยิ่งขึ้น แต่ยังคงความเข้ากันได้ (Compatibility) กับการใช้งาน Kerberos ตามแบบมาตรฐาน นอกจากนี้ ยังสะดวกแก่ผู้บริหารระบบ เนื่องจากไม่จำเป็นต้องแก้ไขใด ๆ ใน Kerberos Authentication Server และ Kerberized Application Server เลย

ผู้จัดจึงหวังเป็นอย่างยิ่งว่า ในอนาคต เราจะพบ Kerberized Application ที่หลากหลายและเป็นประโยชน์แก่ผู้ใช้ แต่ยังคงความปลอดภัยแก่เครือข่ายมากยิ่งขึ้น

5.2 ปัญหาและข้อจำกัดที่พบในการวิจัย

เนื่องจากผู้จัดมีข้อจำกัดในด้านอุปกรณ์ ดังนั้น จึงจำเป็นต้องลดจำนวนอุปกรณ์เพื่อให้เหมาะสม จากเดิม ผู้จัดต้องการจำลองโครงงานให้ Client ขอบริการผ่าน NAT ไปยัง Kerberos Authentication Server และ Kerberized Application Server ใน Internet โดยจำลองโครงงานเหลือเพียง ให้ Client จากเครือข่ายหนึ่งสามารถ access ไปยังอีกเครือข่ายหนึ่งผ่าน NAT เท่านั้น

ข้อจำกัดดังกล่าว ทำให้ผู้จัดมีให้ทดลองการใช้งานที่หลากหลายน้อย อาทิ กรณีการเพิ่มจำนวน extra_address ที่มากกว่าหนึ่งตัวว่ายังสามารถทำงานได้ตามปกติหรือไม่ ซึ่งผู้วิจัยตั้งใจจะทำเสริมฐานความรู้ดิมที่ศึกษาอยู่

และจากวิธีการที่ผู้จัดจึงนำเสนอ ยังพบข้อจำกัดที่ Client จำเป็นต้องทราบถึง IP address ของ NAT ก่อนเพื่อใส่ค่าดังกล่าวใน extra_address ซึ่งอาจมีผู้พัฒนาให้สามารถใส่ค่าดัง

กล่าวโดยอัตโนมัติเพื่อป้องกันไม่ให้ Client ทราบข้อมูลต่าง ๆ ภายในเครือข่ายที่ส่งจากน้าไปใช้ในทางที่ไม่ถูกต้องในอนาคต

5.3 ข้อเสนอแนะในการวิจัยครั้งต่อไป

จากผลงานวิจัยดังกล่าว อาจเป็นพื้นฐานความรู้ในการพัฒนา Kerberized Application ที่มีความซับซ้อนและจำเป็นต้องนำ IP address ไปเป็นส่วนหนึ่งในการให้บริการ อาทิ การทำ Tunnel ระหว่าง Client ที่อยู่ภายใต้ NAT และปลายทางที่อยู่ภายนอก NAT เพื่อทำให้ Client สามารถใช้งาน VPN ที่เป็น Kerberized Application ระหว่างต้นทางและปลายทางได้เป็นต้น

หรืออาจมีการนำความรู้ดังกล่าวไปสร้างเป็นเครื่องมือที่ใช้ตรวจสอบข้อมูลสำหรับผู้ขอบริการที่ Kerberized Application ว่ามาจากที่ใดบ้าง ซึ่งเครื่องมือดังกล่าวอาจเป็นประโยชน์ในการประยุกต์ร่วมกับ CRM เนื่องจาก ผู้ให้บริการสามารถทราบข้อมูลของผู้ใช้บริการว่า มีความถี่ในการขอใช้บริการแค่ไหน ในช่วงเวลาใด และนิยมใช้บริการใดที่สุด เพื่อเป็นแนวทางในการพัฒนาข้อมูลหรือทรัพยากรเครือข่ายให้เหมาะสมต่อไป

แต่ในบางกรณี อาจมีผู้นำแนวความคิดของ Kerberos ในการ compromise ให้ Authentication แบบไม่มี IP address ไปสร้างเป็นเครื่องมือดักจับ Ticket และศึกษาการจำลองงานเป็นผู้ใช้โดยอาศัย Ticket ที่ได้ เพื่อศึกษาถึงจุดบกพร่องของ Kerberos ซึ่งงานดังกล่าว ก็เป็นอีกงานนวัตกรรมหนึ่งที่เป็นประโยชน์ในการพัฒนาแนวทางที่สร้างความปลอดภัยให้แก่ Kerberos ต่อไป

จากการศึกษาและทดลอง ตลอดจนการแสดงผลลัพธ์ที่ได้ ผู้วิจัยหวังเป็นอย่างยิ่งว่า งานวิจัยนี้นั้นคงจะเป็นพื้นฐานความรู้ในการนำ Kerberos ไปประยุกต์ใช้ เพื่อให้มีความสะดวกทั้งต่อผู้ใช้และผู้บริหารระบบ มีความปลอดภัยที่เพียงพอ มีความเป็นส่วนตัว รองรับการทำงานแบบไม่วงไว ขยายระบบต่อในอนาคต และทำให้ข้อมูลในเครือข่ายเป็นหนึ่งเดียวแก่น้ำได้ เพื่อก่อให้เกิดประสิทธิภาพสูงสุดแก่องค์กร

6. เอกสารอ้างอิง

- [1] Randy Chow and Theodore Johnson : Distributed Operating Systems & Algorithm, University of Florida, Addison Wesley, 1997.
- [2] สุรศักดิ์ สงวนพงษ์, สถาปัตยกรรมและโปรโตคอลที่ซีพี/ไอพี, บริษัท ซีเอ็ด จำกัด(มหาชน), กรุงเทพมหานคร, 2543.
- [3] William Stallings, Cryptography and Network Security principles and practices., 3 ed., Prentice Hall, New Jersey, 681p, 2003.
- [4] Elizabeth D. Zwicky, Simon Cooper and D.Brent Chapman, Building Internet Firewalls., 2 ed., O'Reilly & Associates, Inc., CA, 869p, 2000.
- [5] นพดล สาริก, Cryptographic of Information, Individual Studies paper, ภาคเรียนที่ 2, ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี, มหาวิทยาลัยธรรมศาสตร์, 2542.
- [6] Christian Gilmore, David Kormann and Aviel D.Rubin, Secure Remote Access to an Internal Web Server, AT&T Labs -Research, Florham Park, NJ, USA, 1999.
- [7] B. Clifford Neuman and Theodore Ts'o, Kerberos :An Authentication Service for Computer Networks, IEEE Communications, 32(9) ; pp. 33-38., 1994.
- [8] John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so, The Evolution of the Kerberos Authentication System - In Distributed Open Systems, IEEE Computer Society Press ; pp. 78-94, 1994.
- [9] Jennifer G. Steiner, Clifford Neuman and Jeffery I. Schiller, Kerberos: An Authentication Service for Open Network Systems, Massachusetts Institute of Technology, 1988.
- [10] Somesh Jha, Interrealm Authentication in Kerberos Version 5., Computer Science

- Department, University of Wisconsin, Madison, 2003.
- [11] Jonathan Trostle, Irina Kosinovsky and Michael M. Swift, Implementation of Crossrealm Referral Handling in the MIT Kerberos Client, 2001.
- [12] Perter Hernberq, User Authentication How to., <http://www.europe.redhat.com/documentation/howto/user-authentication-howto.html>, 2000.
- [13] IBM Corp, Network Security Enhancements, www.ibm.com/eserver/iseries, 2002.
- [14] Mike Friedman, The UC Berkeley Kerberos Central Authentication Web Server (AWS), ftp://ftp.net.berkeley.edu/kerberos/AWS_V2.doc, Berkeley University, 2003.
- [15] V. Alex Brenen, Kerberos Infrastructure HOWTO version 1.0.3., <http://www.cryptnet.net/fdp/crypto/kerby-infra.html>, 2003.
- [16] Ken Hornstein, Kerberos FAQ v2.0., <http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html#intro.>, 2000.
- [17] NCSA HTTPd Development Team, Kerberos Authentication version 1.5.2a., <http://hoohoo.ncsa.uiuc.edu/docs/index.html.>, 1996.
- [18] Cisco Systems Inc., Cisco IOS Network Address Translation, Copyright 1992-2003.
- [19] Cisco Systems Inc., Network Address Translation and Stateful Inspection in Cisco IOS Firewall for Network security, Copyright 1992-2003.
- [20] Douglas E. Engert, <http://mailman.mit.edu/pipermail/kerberos/2002-June/001144.html>, Argonne National Laboratory, Illinois, 2002.
- [21] Jan De Clercq and Micky Balladelli, Windows 2000 Authentication, Digital, 2001.