



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Security Essentials Bootcamp Style (Security 401)"  
at <http://www.giac.org/registration/gsec>

## Windows 2000 Kerberos Interoperability

GIAC Security Essentials Certification, version 1.4b  
Option 1 – Research on Topics in Information Security  
Christopher Nebergall  
November 27, 2002

### Abstract

While other papers discuss Kerberos in general or the Windows 2000 implementation, this work explores compatibility issues between traditional Unix implementations and Microsoft's implementation. First discussed will be Microsoft's support of the official Kerberos V5 standard RFC 1510. Next discussed will be how to configure a Windows 2000 network to work with a UNIX Kerberos implementation in a variety of common scenarios, and finally this work discusses extensions to the Kerberos standard that Microsoft has documented in a number of Internet Drafts. Overall Microsoft's Kerberos implementation is proving itself to be interoperable in all common scenarios.

### Windows 2000 Kerberos

With the launch of Windows 2000 Microsoft began supporting Kerberos V5 as its default authentication protocol. Many people praised Microsoft's adoption of Kerberos because the protocol has proven itself to be a secure and efficient authentication algorithm on other platforms for a number of years. It also raised the hopes of many people that Microsoft's Windows authentication architecture would finally begin to interoperate with other operating systems, allowing system administrators to simplify the administration of accounts that previously had to be managed separately from each other.

### Standards Support

#### *Encryption Types*

Microsoft officially supports 128 bit RC4-HMAC as its default encryption type for Kerberos tickets but also supports DES-CBC-CRC and DES-CBC-MD5 encryption to maintain compatibility with MIT Kerberos. A table of the key sizes is included below.

<b>Encryption types</b>	<b>Authentication (Key Length in bits)</b>	<b>Signing (Key Length in bits)</b>	<b>Privacy (Key Length in bits)</b>
DES-CBC-CRC	56	56	56

DES-CBC-MD5	56	56	56
RC4-HMAC	128	128	56 (128 w/ the High Encryption Pack installed)

Microsoft documents its implementation of the RC4 –HMAC algorithm and supported key lengths in Swift and Breznak's [Internet Draft](#). Microsoft choose RC4 as its default encryption type because the algorithm was approved for export early in the development of Windows 2000 before export of DES had been approved. Additionally, NT 4 already supported RC4 so the appropriate keying material was available in accounts upgraded from NT 4 without requiring users to change their passwords. Unfortunately, Microsoft chose not to support 168 bit 3-DES, which according to Microsoft's document was (is?) still controlled by export regulations.

If users don't wish to downgrade to 56-bit DES encryption to interoperate with Microsoft they can wait for the 1.3 release of MIT Kerberos which according to discussions on the Kerberos\_V5\_Development mailing list is scheduled to have RC4 support, or they can instead use Heimdal Kerberos (<http://www.pdc.kth.se/heimdal/>), which already supports RC4.

If users' accounts were upgraded from NT 4 in order to use DES encryption the users must change their password before the appropriate keying material becomes available. Additionally, the Windows 2000 network administrator account is initially created without a DES key so the owner of this account must also change his or her password before DES encryption can be used.

#### *Ticket types*

Microsoft has no support for post-dated or proxy tickets. Microsoft does offer the use of address-less TGT's, which in many environments is more convenient (through NAT for example) but may be considered a security risk by some administrators.

#### *Programming API's*

Microsoft does not support the GSSAPI (RFC 2078) for Win32 programmers but instead supports what they call the Security Service Provider Interface (SSPI). Software written to the Kerberos SSPI is "wire compatible" with GSSAPI implementations of Kerberos (and therefore compatible with RFC 1964). Microsoft has an [MSDN article](#) summarizing the difference between the API implementations. Like the GSS-API the SSPI supports delegation, authentication, and message integrity and privacy. The table below includes some of the GSS function calls and their equivalent SSPI function calls.

GSSAPI Function	SSPI Function
GSS_init_sec_context	InitializeSecurityContext
GSS_get_mic	MakeSignature
GSS_verify_mic	VerifySignature
GSS_Wrap	EncryptMessage
GSS_Unwrap	DecryptMessage

If an SSPI-based client requests delegation to an MIT or SSPI-based service two Microsoft specific requirements must be met before a TGT will be delegated.

1. The user account in Active Directory of the user running the client application cannot have the *"This account is sensitive and cannot be delegated"* property set.
2. The service ticket for the service principal must have the *OK\_AS\_DELEGATE* flag set.

The *OK\_AS\_DELEGATE* flag was added to the latest Kerberos V5 Internet Draft designed to update RFC 1510. This flag provides information to the user from the KDC specifying whether a particular service principal is trusted to accept delegated credentials. If the service is running with system privileges and on a computer account registered in Active Directory simply check the box *"This computer is trusted for Delegation"* on the machine's account in **Active Directory Users and Computers**. If the Kerberos service is running under the rights of a user (such as a Kerberos service on a Unix machine) check the *"Account is Trusted for Delegation"* property on the user account. See the interoperability scenarios section below for more information on how to deal with a KDC that does not support this flag.

Microsoft also does not support the raw krb5 API, but does make an extra API available (*LsaCallAuthentication*) for accessing the ticket cache.

### *Principal names*

One incompatibility -- which arguably deviates from the RFC 1510 standard -- is Microsoft's case insensitive principal names. All possible alphabetic cases of a principal name ([username@realm](#)) are equivalent to Microsoft Active Directory, and they all map to the same network account. Traditionally, Windows log-on names have been case insensitive so it should be of no surprise that Microsoft has chosen to continue this pattern with their Kerberos' support. Unfortunately, this practice will cause problems with applications designed to work with traditional Kerberos implementations that expect only one possible representation of a principal name.

The biggest problems occur when users are allowed to choose the case of their principal name by how they log in to the network. For example, in the two scenarios discussed below if a user logs in as "joe\_user" he or she will receive a ticket for [joe\\_user@DOMAIN.COM](#) (all lower case username), but if a user logs in as JOE\_USER they will receive a ticket for the principal [JOE\\_USER@DOMAIN.COM](#) (all uppercase username).

Causes:

1. If the user has *the "Use DES Encryption Only"* attribute checked on his or her user account in Active Directory.

or

2. If the user's AS request is from a traditional MIT or Heimdal Kerberos client.

The exact cause of the second case is undocumented; somehow Microsoft is able to differentiate between AS requests from their own Windows clients and non-MS Kerberos clients. A useful area of research would be to figure out exactly how Microsoft is able to differentiate between the client types.

This behavior will cause confusion because users will be able to authenticate to traditional Kerberos services, but their request might still fail because the service will not be able to recognize their principal name when it attempts to do its own local authorization checks. There are no known workarounds short of a) training users to always log in using the same case or b.) re-working applications designed around traditional Kerberos to do only case-insensitive compares between principal names. Neither option is ideal.

## **Interoperability Scenarios**

In Microsoft's "Answers to Frequently Asked Kerberos Questions" page they claim to have successfully completed compatibility testing with Kerberos implementations developed by MIT, Heimdal, CyberSafe, IBM and Sun. Listed below are instructions on how to set up several of the interoperability scenarios described by Microsoft including how to set up a trust relationship between realms, how to give Windows users access to Kerberos services, how to map a Kerberos user account to an Active Directory user account, and finally how to register Kerberos hosts and services in Active Directory.

### *Trust Relationships*

A Microsoft Active Directory Network can be configured to trust a UNIX Kerberos KDC. In order to set up a trust relationship the address of the UNIX Kerberos KDC and realm must be registered with the domain controller. Next, the password to use for the shared key must be configured. The exact steps for setting up the default

non-transitive trust relationship are included below for the imaginary MIT Kerberos realm MITREALM.DOMAIN.COM and the Microsoft Window's domain DOMAIN.COM. To set up a transitive trust relationship between the local network and the MIT Kerberos Realm the netdom tool can be used which is provided with the Windows 2000 Resource Kit.

1. Run the following from the command prompt of the domain controller. (Note: kdc.mitrealm.domain.com is the hostname of the MIT Kerberos KDC.)

```
Ksetup /addkdc MITREALM.DOMAIN.COM kdc.mitrealm.domain.com
```

2. On the start bar choose **Programs->Administrative Tools-> Active Directory Domains and Trusts**
3. Click on properties of MITREALM.DOMAIN.COM select the *Trust tab* and click *Add*.
4. Enter a password that will be used to create the shared key between the domains.
5. When asked whether this is a non-windows domain click **OK**.

Now the trust relationship must be configured on the Unix Kerberos KDC.

1. *kadmin -q "ank -pw password krbtgt/DOMAIN.COM@MITREALM.DOMAIN.COM"*
2. *kadmin -q "ank -pw password krbtgt/MITREALM.DOMAIN.COM@DOMAIN.COM"*

#### *Allowing Windows Workstations access to Services on a non-Microsoft Kerberos Realm*

For a Windows Workstation to have access to Kerberos services on a non-Microsoft Kerberos realm each workstation needs to know of the location of the KDC for that realm.

Run the following command on each Windows workstation.

```
Ksetup /addkdc MITREALM.DOMAIN.COM kdc.mitrealm.domain.com
```

If users need the ability to delegate to services in non-Microsoft realms these realms must provide the OK\_TO\_DELEGATE flag in the service tickets they provide, or Microsoft gives the option of setting a registry value which simulates turning on this flag for every service in a particular realm. To enable this feature, edit the realmflags attribute at the registry location listed below and ensure that the third least significant bit is set (value 4 if you do not need any other options enabled).

HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Domains\<Realm Name>

See the Microsoft Technical Reference for the Windows 2000 Registry for more information on Kerberos registry values.

### *Mapping Kerberos User Principals to Active Directory User Accounts*

In order for a non-Windows Kerberos user to access resources on a Microsoft network, there must be a trust between the user's KDC and Microsoft Active Directory and a mapping between the user's non-Windows principal name and an Active Directory user account. This mapping is required so that Microsoft can add the required authorization information into the user's tickets for use on the Windows 2000 Network. Instructions on how to achieve the mapping between the Kerberos principal JoeUser@MITREALM.DOMAIN.COM and AD user account JoeUser are included below.

1. Create an account for the user in Active Directory using **Active Directory Users and Computers** for example JoeUser
2. Check the **Advanced** menu item under the **View** menu.
3. Right click the JoeUser Account and select **Name Mappings**.
4. Select the **Kerberos Mappings** tab.
5. Add the complete principal name of the user from the Kerberos realm. For example JoeUser@MITREALM.DOMAIN.COM.

### *Registering a Unix Kerberos Service in Active Directory*

If a Windows 2000 user wishes to authenticate to a Kerberos service on non-Windows machine that service must be registered in Active Directory. The following example maps the Unix service http/saturn.domain.com to the Active Directory user account saturn\_server in the domain DOMAIN.COM

1. Create a user account for saturn\_server using **Active Directory Users and Computers**.
2. Type the following from the command line

```
C:> ktpass -princ http/saturn.domain.com@DOMAIN.COM -mapuser saturn_server -pass password -out keytabfile
```

3. Copy the keytabfile to the Unix machine and merge it with the systems keytab file.

### *Registering a Unix Host in Active Directory*

The following example maps the Unix host saturn.domain.com to the Active Directory user account saturn in the domain DOMAIN.COM.

1. Create a user account for *saturn* using **Active Directory Users and Computers**.
2. Type the following from the command line

```
C:> ktpass -princ host/saturn.domain.com@DOMAIN.COM -mapuser saturn -pass password -out keytabfile
```
3. Copy the keytabfile to the Unix machine and merge it with the systems keytab file.

### *Configuring a non-Windows host to use Active Directory as its KDC*

Kerberos clients on a non-Windows machine can be configured to use Active Directory as their default KDC. The name of the Windows domain in all uppercase is the name of the Windows realm. For the example below the windows domain DOMAIN.COM has the machine *server.domain.com* configured as a domain controller.

1. Set the default realm.

```
[libdefaults]
default_realm = DOMAIN.COM
```

2. Optionally set the encryption types.

```
default_tkt_enctypes = des-cbc-crc des-cbc-md5
default_tgs_enctypes = des-cbc-crc des-cbc-md5
```

3. In the realms section set the hostname of the kdc.

```
[realms]
DOMAIN.COM = {
  kdc = server.domain.com:88
}
```

4. Optionally map the hostname to the name of the realm.

```
[domain_realm]
.domain.com = DOMAIN.COM
```

### **Kerberos Extensions**

Microsoft has made a number of extensions to the Kerberos protocol to extend the abilities of Kerberos for the Microsoft environment. Microsoft should be commended for referencing the documents they use for these extensions and writing their own documents when such standards did not previously exist.



## PKINIT

Microsoft supports Smart Card Authentication to the Kerberos KDC using draft 9 of IETF's "[Public Key Cryptography for Initial Authentication in Kerberos](#)" (PKINIT) Internet Draft. Unfortunately, Draft 9 has long since expired and current drafts (Draft 16 as of this writing) have changed since earlier revisions. The standard itself mandates that Diffie Hellman keys are supported in addition to Digital Certificates, but Microsoft did not implement this part of the standard. The major steps of Microsoft's implementation are included below.

1. In the preauthentication field of the AS request (the initial request used to get a ticket granting ticket) a user sends his or her certificate and a unique authenticator signed with the certificate's corresponding private key.
2. The KDC verifies that the certificate and the signature are valid.
3. The KDC creates an AS reply with the encrypted part of the reply encrypted using the user's public key and signed using the KDC's private key.

The Meta Center (<http://meta.cesnet.cz/software/heimdal/pkinit.en.html>) has some early patches available for Heimdal to enable basic PKINIT support.

### *Privilege Attribute Certificates*

Microsoft received a lot of attention for including authorization data in Kerberos tickets. Many thought it went against the spirit of the Kerberos standard to include authorization data in this fashion. This information, called the Privilege Attribute Certificate or PAC, is encoded in the authorization field of tickets generated by Microsoft domain controllers. It includes a list of groups for which the ticket's owner belongs. This field is normally ignored by non-Microsoft services, but Microsoft has released their specification for how the authorization data is encoded, so non-MS Kerberos services could take advantage of the privilege information. See the document "[Utilizing the Windows 2000 Authorization Data in Kerberos Tickets for Access Control to Resources](#)" in the references section for more information.

### *Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols ([RFC 3244](#))*

Microsoft added an extension to the Kerberos set password protocol to allow an authorized principal to change the password of another principal. This is necessary to allow an administrator to set passwords for other network users. Additional changes included the removal of the requirement for the service ticket to have the initial flag set and an updated protocol version number. Microsoft also supports the original change password protocol used by MIT Kerberos.

## *DNS Service Records*

Microsoft supports locating both the KDC and the Kerberos password changing service through DNS service records (`_kerberos` and `_kpasswd` respectively). MIT Kerberos also enabled default support for locating the local KDC through service records in version 1.2. MIT does not enable by default the ability to look up the realm of host because of possible DNS spoofing issues.

## *User-to-User Kerberos Authentication*

Microsoft Windows implements a SSPI version of the Kerberos User-to-User protocol specified in the "[User to User Kerberos Authentication Using GSS-API](#)" Internet Draft. The protocol is designed for cases when it would be considered unsafe for users' to leave their long-term credentials vulnerable while they host a short-term service. This protocol differs from the normal protocol because instead of using the long-term key of the service (in this case a temporary service set up by another user) for communicating the session key, the key of the recipient's TGT is used.

MIT currently does not support the User-to-User protocol through the GSSAPI interface but does support the protocol using the pure krb5 API.

## *KDC Referral Mechanism*

Microsoft supports extensions to the KDC referral mechanism according to the "[Generating KDC Referrals to locate Kerberos realms](#)" Internet Draft. The current mechanism requires the client to be configured with information about every realm for which it has access. This Internet Draft describes extensions to Kerberos that would allow the user's KDC to refer the user to the correct KDC or service, therefore reducing the amount of client configuration required. The following new types of referrals are introduced:

1. AS Ticket Referrals- Referral used when a user does not know for which realm they wish to authenticate.
2. TGS Ticket Referrals- The user does not know the realm of the service he or she wishes to access.
3. Cross-Realm Shortcut referrals- The KDC chooses the next realm the user will access in the referral chain.

## **Conclusion**

Microsoft's Kerberos implementation is proving itself to be very interoperable with other Kerberos implementations. The one remaining difficulty is Microsoft's case-insensitive principal names and this area is survivable with some difficulty. Areas for future study would be to investigate the feasibility of completely replacing a UNIX KDC with Microsoft AD and how well Microsoft's domain controllers could handle the

increased load. Furthermore, while in limited test cases Microsoft's Kerberos has proved interoperable, a real world example of how an organization integrated their Active Directory and traditional Kerberos authentication architectures would prove to be invaluable.

## Sources

Brezak, John. "Utilizing the Windows 2000 Authorization Data in Kerberos Tickets for Access Control to Resources."

Feb. 2002. URL: [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnkerb/html/MSDN\\_PAC.asp?frame=true&hidetoc=true](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnkerb/html/MSDN_PAC.asp?frame=true&hidetoc=true) (19 Nov. 2002).

De Clercq, Jan. Ballardelli, Micky. "Windows 2000 Authentication." Mission-Critical Active Directory: Architecting a Secure and Scalable Infrastructure. Digital Press, March 2001, URL: <http://www.windowstlibrary.com/Content/617/06/6.html> (19 Nov. 2002)

"Kerberos 5 Release 1.2." MIT Kerberos Release Notes. 15 Nov. 2002. URL: <http://web.mit.edu/kerberos/www/krb5-1.2/index.html> (21 Nov. 2002).

"Microsoft Knowledge Base Article – 230669: Windows 2000 Kerberos 5 Ticket Flags and KDC Options for AS\_REQ and TGS\_REQ Messages." 10 Oct. 2002. URL: <http://support.microsoft.com/default.aspx?scid=kb;en-us;230669> (19 Nov. 2002)

"Microsoft Knowledge Base Article – 248808: Administrator Account Is Not Usable by Non-Windows 2000 Kerberos Clients." 9 Nov. 2002. URL: <http://support.microsoft.com/default.aspx?scid=KB;en-us;248808&> (21 Nov. 2002)

"Microsoft Knowledge Base Article – 266080: Answers to Frequently Asked Kerberos Questions", 11 Oct. 2002. URL: <http://support.microsoft.com/default.aspx?scid=KB;en-us;q266080> (19 Nov. 2002)

Neuman, Clifford, et al. "The Kerberos Network Authentication Service (V5)." 1 Nov. 2002. URL: <http://www.ietf.org/internet-drafts/draft-ietf-krb-wg-kerberos-clarifications-02.txt> (19 Nov. 2002)

Neuman, Clifford, et al. "Public Key Cryptography for Initial Authentication in Kerberos." Draft 9. URL: <http://www.globecom.net/ietf/draft/draft-ietf-cat-kerberos-pk-init-09.html> (Nov. 22, 2002).

“SSPI/Kerberos Interoperability with GSSAPI.”

URL: [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/sspi\\_kerberos\\_interoperability\\_with\\_gssapi.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/sspi_kerberos_interoperability_with_gssapi.asp) (21 Nov. 2002).

“Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability.” Jan. 10, 2000.

URL:

<http://www.microsoft.com/windows2000/techinfo/planning/security/kerbsteps.asp> (19 Nov. 2002).

Swift, M. Brezak J. "The Microsoft Windows 2000 RC4-HMAC Kerberos encryption type." Draft 4. May 2002.

URL: <http://www.globecom.net/ietf/draft/draft-brezak-win2k-krb-rc4-hmac-04.txt> (19 Nov. 2002).

M. Swift. “Generating KDC Referrals to locate Kerberos realms.” Oct. 1999 URL: <http://www.globecom.net/ietf/draft/draft-swift-win2k-krb-referrals-00.html> (26 Nov. 2002).

Swift, M. et al. "Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols." Feb. 2002. URL: <http://www.ietf.org/rfc/rfc3244.txt> (19 Nov. 2002)

Swift, M. et al. “User to User Kerberos Authentication using GSS-API.” Draft 3. Oct. 2001

URL: <http://www.globecom.net/ietf/draft/draft-swift-win2k-krb-user2user-03.html> (26 Nov. 2002)

Technical Reference to the Windows 2000 Registry (Windows Help File regentry.chm): Windows 2000 Resource Kit. Copyright Microsoft Corporation 1995-2000.

© SANS Institute 2003. Author retains full rights.

# Upcoming Training

Click Here to  
**{Get CERTIFIED!}**



SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor