Taming the beast : Assess Kerberos-protected networks

[Work in progress – Black Hat EU 2009]

Emmanuel Bouillon

Commissariat à l'Energie Atomique, Centre DAM-Île de France, Bruyères-le-Châtel
91297 Arpajon Cedex, France
emmanuel.bouillon@cea.fr

Abstract. Due to its universal support, to the fact that it is Microsoft's default and that it provides for a real SSO solution, Kerberos is a pervasive authentication protocol with a strong reputation of security. This talk will cover some of the issues involved with assessing a Kerberized network both under Unix and Microsoft Windows environment. It will review known yet underestimated implementation limitations and study under which circumstances they still lead to exploitable vulnerabilities. It will also present new ones that enable to step in the targeted systems. Finally, we will discuss some of the protocol evolutions and study their potential consequences in terms of security.

Keywords: Kerberos, security, replay attacks.

1 Introduction

Kerberos is a sophisticated network authentication system, that has been publicly available since 1989 and that provides for the eternal holy grail of system administrators: a secure single sign on solution.
While many large organizations and academic institutions have enjoyed the benefits of using Kerberos in their network, the deployment of Kerberos met a tremendous growth when adopted by Microsoft as its default authentication mechanism in Active Directory (within Microsoft Windows 2000). Unfortunately, this "transparent" support made some system administrators not really aware of the fact they use Kerberos on a daily basis and that they actually manage a KDC (Key Distribution Center – the Kerberos authentication server).

Due to its universal support, to the fact that it is Microsoft's default[1] and that it provides for a real SSO solution, Kerberos is a pervasive authentication protocol with a strong reputation of security. Very probably this is the most used authentication protocol for end users on their company LAN.

However, lots of system administrators still make dramatic mistakes while configuring it (those mistakes are made more likely by buggy GUIs and their poor documentation) and few pen-testers really know how to assess such kind of environments.
This talk will help those two kinds of people to better do their job when coping with Kerberos.
Furthermore it will demonstrate that there still exist underestimated and unknown implementation vulnerabilities that need to be addressed. Finally we will discuss new perspectives offered by some of the recent Kerberos protocol's evolutions.

1.1 Agenda

This presentation will be divided into four main parts:

- A quick recap of the Kerberos protocol. So that everyone will have in mind the key points of this protocol.
- An introduction of two (supposedly) known attacks on Kerberos authentication mechanism
  - One is called KDCspoofing
  - The other one is referred to as "replay" attacks
- Discussion on how a combination of those two previous attacks can lead to unexpected security flaws.
- TGT harvesting and which conditions make it possible. Protocol's evolutions and their potential impacts on security will be mentioned.

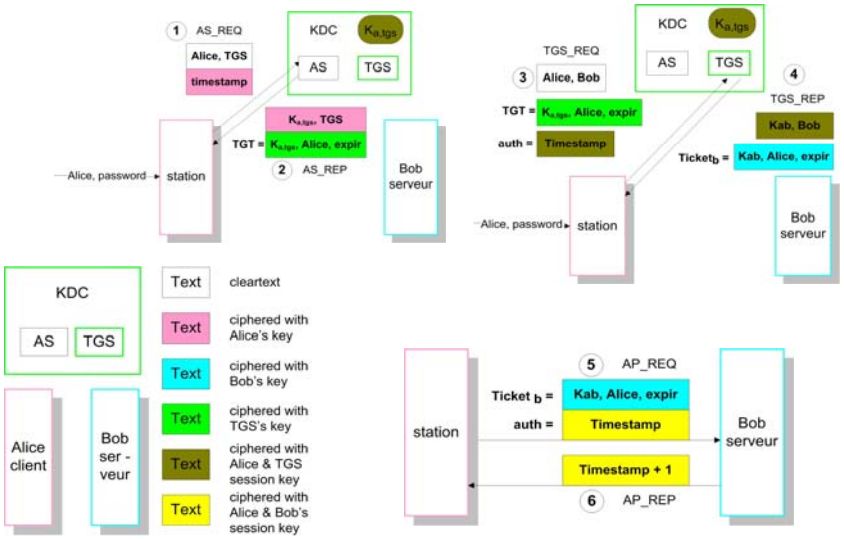
1 For computers that are members of Windows domains

2. Kerberos in a nutshell

A Kerberos authentication needs three pairs of request/response exchange :

- 1st pair : AS-REQ/AS-REP (for authentication server request and response) is the one that allows a user to obtain a TGT (Ticket Granting Ticket). This ticket can be seen as a “generic” ticket obtained by a user once authenticated by the KDC and that allows then the user to get tickets for services (called TS for Ticket Service) without having to use its password again. This exchange occurs between the client and the KDC and requires the knowledge of the originating principal secret shared key (the password for a user). Note that every ticket has a limited lifetime (typically several hours for a TGT).
- 2nd pair : TGS-REQ/TGS-REP (for Ticket Granting Server request and response) is the one that occurs between the client and the KDC when the client wants to obtain a service ticket for a given service. Note that the client does not use its password in order to get a TS.
- The last exchange, AP-REQ/AP-REP (for Application Request and response) occurs for mutual authentication between the client and the accessed service. Therefore this exchange occurs between the client and the service's server (no KDC).

Highly simplified description of Kerberos authentication :



At the end of the first step, both the user and TGS share a session key and after the last step the user and the server are mutually authenticated and share a session key. Step 2 and 3 can be renewed several times to get access to several servers without using the user's password as long as the TGT remains valid (SSO).

3. KDC spoofing and Replay attack

1. KDC spoofing

This refers to an attack which relies basically on the ability to spoof KDC responses. Having in mind the Kerberos protocol description, spoofing KDC response should not be a security concern. Indeed, Kerberos has been design to bear an untrusted network. IP spoofing is something that happens on untrusted networks. Kerberos protocol performs mutual authentication. End user's and server's identities need to be proven. This ensures protection against Man-in-the-Middle attacks. Yet circumstances still exit under which this might represent a real risk. For instance, some applications under Unix systems such as PAM modules available for authentication against Kerberos passwords do not use the whole Kerberos authentication process by default. Instead they stop after the first AS-REQ/AS-REP exchange. They use a “shortcut” : send an AS-REQ and try to decrypt the AS-REP using the provided password (step 1,2 of the schema). In case of success, the PAM module returns PAM_SUCCESS. The correct behavior is to validate the TGT asking for a TS for the local system principal and verifying it using the local keytab file (step 3,4,5,6). This shortcut opens the door the the KDC spoofing attack made popular by Dug Song's code.
A properly configured Kerberos PAM module solves this problem. Yet, we see two concerns in that matter. When assessing such kind of environments we still find misconfiguration regarding this vulnerability. This mistake are made more likely by buggy “Kerberos in two clicks” GUIs and their poor documentation. The other deals with the fact that to circumvent KDC spoofing, it is required to validate TS for the local system principal (usually host/machine.domain). To do so, the authenticating process needs this principal secret key, usually stored in a file called “keytab”. For security reasons, this file needs to be readable only by root. Unfortunately, some “pamified” processes do not have root privileges. For instance, every lock screen systems or applications like gnu screen. Missing this readable privilege on the keytab file, these applications cannot verify user's TGT and cannot go through the whole Kerberos authentication. The behavior in that case is to allow access based on the ability to decipher the AS-REP using the provided password. This fail-unsafe default means that these applications remain vulnerable to KDC spoofing attacks.

2. Replay attack

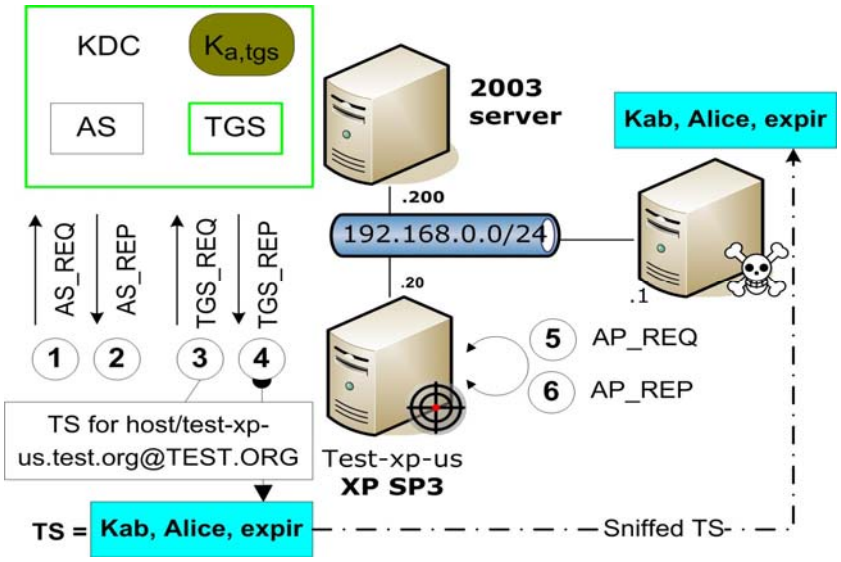
This attack consists in reusing a previously sniffed AP-REQ and in order to impersonate the legitimate user against the accessed service. As with the KDC spoofing attack, the replay attack requires the ability to listen to network messages as well as the ability to send fake ones. Several mechanisms are in place in order to make this attack more difficult or impossible.

- Timestamps : Time-based authenticators shrink the time window during which the authenticator can be reused.
- Ticket can be address-full, meaning that IP addresses for which tickets have been generated get embedded inside the ticket. Thus the server is able to verify these addresses against the connection source IP address. While it might represent a true challenge to enforce address-full tickets in a realistic environment, very few common services actually verify these addresses, in fact none in a Microsoft Windows environment and only KDC services under Unix environment.
- Replay cache : a server can store previously submit authenticators during their lifetime and detect their reuse.
- The last countermeasure is to use keyed cryptographic checksum in upper layer protocol using the session key (unknown by the replaying attacker).

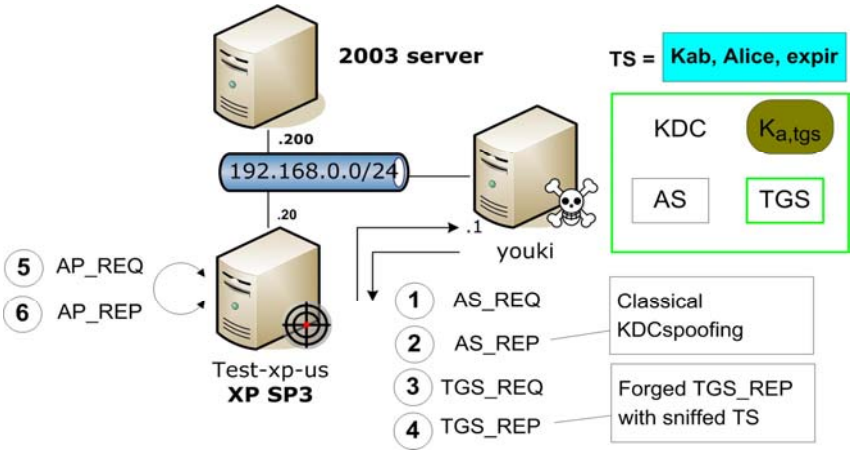
4. "Pass the ticket"

Knowing the mechanisms involved in the two previously described classical attacks, it is possible to combine their mechanisms in order to impersonate a user on a Windows XP machine and try to thwart Windows "anti-KDC-spoofing" mechanism. This Man in the Middle attack uses KDC spoofing techniques to generate the AS-REP and replay techniques but not replaying AS-REP responses but TGS-REP responses.

Step 1 : Sniff legitimate connection



Step 2 : KDCspoof + Replay



It seems that the identity used by the workstation is the one included in the Privilege Attribute Certificate (PAC) part of the TGS-REP message and ciphered by the workstation service key. PAC is a key point of the Microsoft's Kerberos implementation and will be discussed later in this presentation. This allow an attacker to impersonate a user if one valid connection for that user has been previously sniffed.

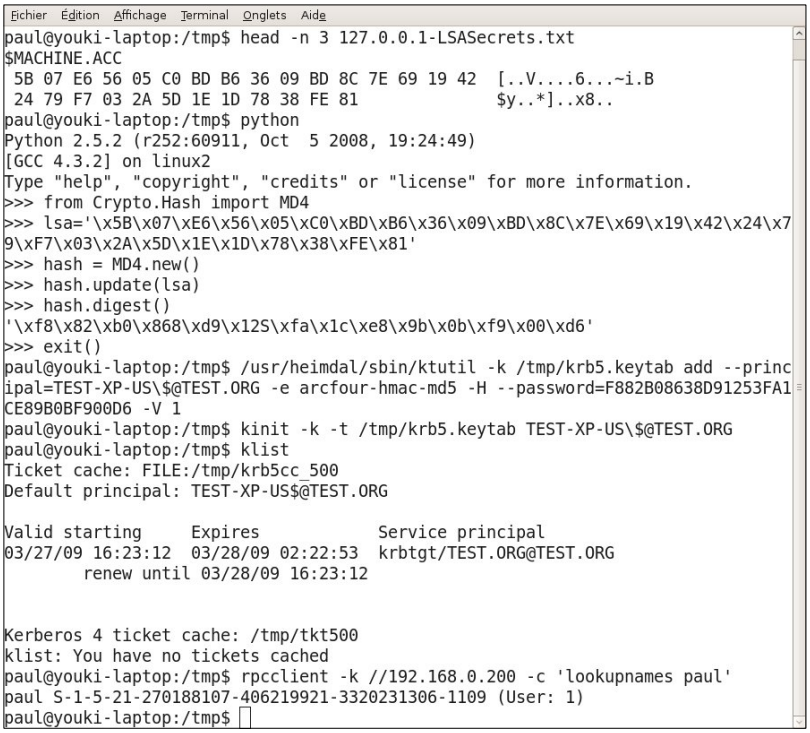
The range of this attack is the local LAN and it requires a sniffed TGS-REP for the targeted user and workstation and the ability to send fake packets on the network. Kerberos redirection flow could be achieved beyond the LAN if features like DNS dynamic updates are allowed.

It is also possible to trigger TGS-REQ for a given service providing the fact that this server is seen by the targeted user as part of his or her Local Intranet zone. For instance by forcing the targeted user to connect to a Web server using SPNEGO negotiation for authentication, this server being in the same DNS domain as the targeted computer. Indeed by default, using Internet Explorer, if the computer name portion of the requested URL does not contain periods (ex: http://server), the server is consider to be on the Local Intranet zone. In this case, IE Integrated Windows Authentication will trigger a TGS-REQ for the service HTTP/server.

5. Users impersonation

1. Compromise of a user's credentials

On Unix environment, tickets are stored in temporary files only readable by there owners. Root local escalation or user's environment files tampering can make an attacker able to steal these credentials and so, impersonate the targeted user. The point is that even in the case of address-full tickets, once stolen they can usually be reuse on other machines. In fact only the TGT cannot be reused from other IP address. Valid



Service Tickets, even including address filed, can be leverage (as long as they remain valid).

This emphasizes the importance of the choice of tickets lifetime et renewable time.

2. Root/system compromise of a client machine

On Unix environment, from a Kerberos perspective, a root compromise of a client machine means access to any ticket caches and to the keytab file. With this file, it is possible to forge Service Tickets for any users towards the service principals contained in the keytab file. It is impossible to impersonate every valid user on this machine.

Under Windows environment, this is the same with two details :

- There is no keytab file. The secret key of the machine's principal can be obtained from the LSA secret more precisely from the $MACHINE.ACC key.

The same machine's key can be gained thanks to whosthere.exe.

```
  C:\Work\pshtoolkit_v1.4\whosthere>whosthere.exe

  WHOSTHERE v1.4 - by Hernan Ochoa (hochoa@coresecurity.com,
hernan@gmail.com) - (
c) 2007-2008 Core Security Technologies

  This tool lists the active LSA logon sessions with NTLM credentials.
  (use -h for
 help).

  -B is now used by default. Trying to find correct addresses..Found!.

  the output format is: username:domain:lmhash:nthash


  Administrator:TEST-XP US:
8AAB1F7CD4F792A1EAD3B435B51404EE:17952F051CEF3C8A55A341
DFACFD86DE

TEST-XP-US$:TEST:
00000000000000000000000000000000:F882B08638D91253FA1CE89B0BF900
D6
```

The knowledge of this kind of key can help pen-testers to be part of the domain and start the resources and users enumeration (SIDs, shares, ...).

- In order to impersonate a user connecting to a machine when knowing its Kerberos secret key you need to forge a user's PAC (Privilege Attribute Certificate). PAC is an extension to the Kerberos protocol within Microsoft's implementation. It contains information like user's SID as well as a list of groups to which the user belongs. A PAC includes two keyed checksums one using the server's secret key and the other one using the KDC key (krbtgt) itself. Only the first one can be check by the server and so, only the first one needs to be forged using the previously obtained server's key.
  This way, as for a Unix system when knowing its keytab, it is possible to impersonate any user on a Windows XP machine when knowing its $MACHINE.ACC key purely base on Kerberos authentication protocol.

3. TGT harvesting

From a pen-tester perspective, an interesting target is users' (if possible with privileges) TGT. This allows full impersonation during the TGT's lifetime, typically several hours.
Under Unix environment, this quite always means stealing temporary files containing credentials, either by tampering user's environment, browsing /tmp directory on owned machines, or get a targeted user to connect (e.g. ssh) to a owned machine and leverage ticket's forwarding. Therefore ticket forwarding policy is a key point to

consider when assessing a kerberized environment. Especially in the case of one-way trust relationship between to realms.
Windows environment usually make TGT harvesting more difficult since credentials cache are not stored on temporary files and that default options (OK_AS_DELEGATE for principals and AllowTGTSessionKey registry key). For instance, IE won't allow a TGT to be forwarded to a server, even located in the Intranet Local zone, if this principal has not the OK_AS_DELEGATE flag. That is why getting the principal's secret key o such a principal is extremely interesting when trying to steal users' TGTs. The granularity of this protection has been improved by the extension “Service for User” and “Constrained Delegation”.

Conclusion

Kerberos provides a secure, scalable, cross-platform, open SSO solution. This authentication protocol plays a key role in distributed domain security in Windows. Yet complex, there is still a need to raise awareness in order to achieve better security. This talk is an effort to raise Kerberos security awareness and provides security and system administrators with information for that purpose. Only a few points have been discussed here. Other subtleties need to be checked when auditing a Kerberos network. Especially there are many practical issues to be considered (pre-authentication, keytab deployment procedures, unattended/non interactive service connections, ticket life and renewal times, crypto-system of cross-realm keys, etc.).

References

1. S. M. Bellovin, M. Merritt : Limitations of the Kerberos Protocol, Winter 1991 USENIX Conference Proceedings
2. http://monkey.org/~dugsong/kdcspoof.tar.gz
3. Joel Scambray, Stuart McClure : Hacking Exposed - Windows, 3rd Edition, ISBN 978-0-07-149426-7
4. C. Neuman, T. Yu, S. Hartman, K. Raeburn: RFC 4120 - The Kerberos Network Authentication Service (V5)
5. Kevin Johnson, Ed Skoudis, Joshua Wright – InGuardians

- The Pen Test Perfect Storm – Part I

6. Privilege Attribute Certificate Data Structure, http://msdn.microsoft.com/en-us/library/cc237917(PROT.10).aspx
7. Brian Tung : Kerberos – A Network Authentication System – Addison-Wesley – ISBN 0-201-37924-4
8. Jason Garman : Kerberos: The Definitive Guide – O'Reilly - ISBN 10: 0-596-00403-6
9. Kimmo Kasslin, Antti Tikkanen : Attacks on Kerberos V in a Windows 2000 Environment
10. Kimmo Kasslin, Antti Tikkanen : Replay Attack on Kerberos V and SMB

11. Kimmo Kasslin, Antti Tikkanen and Teemupekka Virtanen : Kerberos V Security: Replay Attacks
12. H.D. Moore, Valsmith : Tactical Exploitation
13. Mark E. Russinovich, David A. Solomon, MS Windows Internals - 4th Edition – ISBN 13: 978-0-7356-1917-3