

Attacks on Kerberos V in a Windows 2000 Environment

Kimmo Kasslin
kimmo.kasslin@hut.fi

Antti Tikkanen
antti.tikkanen@hut.fi

Abstract

Kerberos V is a trusted third-party authentication mechanism designed for TCP/IP networks. It uses strong symmetric cryptography to enable authentication in an insecure network. Microsoft introduced Kerberos V as the authentication mechanism for Windows 2000. It is used in many networking applications. An example is SMB, which is a protocol used for file and print services. In this paper we discuss attacks against Kerberos V that enable retrieving passwords and stealing users' identities on the local network. SMB is used as an example in one of the attacks. We also discuss Windows 2000 implementation specifics that affect the feasibility of these attacks.

1 Introduction

Kerberos was developed at MIT as a part of Project Athena. It is based on a key distribution model invented by Roger Needham and Michael Schroeder. Symmetric cryptography and a trusted third-party are the basis of this authentication mechanism. There have been two versions of the protocol in public use, namely Kerberos IV and V. In this paper we discuss only Kerberos V, which has multiple advantages over the previous version.

Kerberos V is the authentication mechanism used in Windows 2000. It is used to authenticate users logging into workstations on a domain environment and to other network services. In this paper we use SMB (Server Message Block) as an example of a protocol that primarily uses Kerberos for authentication in a Windows 2000 domain. SMB is the protocol used for file and print services.

The security of Kerberos has been discussed in several papers: see [1] for an example. Possible weak points include password attacks against Ticket-Granting tickets or pre-authentication data, replay attacks, attacks against network time protocols (Kerberos requires time synchronization) and malicious client software. In this paper, we focus on the first two scenarios: password attacks and replay attacks. We show that a password attack is feasible, thus allowing the attacker to discover weak user passwords. We use pre-authentication data for this attack. A replay attack is presented with the SMB protocol. This allows an attacker to access file shares with the victim's credentials without actually knowing the password.

The chapters are divided as follows: Chapter 2 includes technical descriptions of the protocols discussed in this paper. Chapter 3 will cover some of the vulnerabilities in the Kerberos V protocol. We discuss the attacks we implemented in chapter 4, and analyze the results of these attacks in chapter 5. Some possible protection mechanisms are described in chapter 6. Finally, in chapter 7, we draw conclusions from the presented results and discuss possible future research.

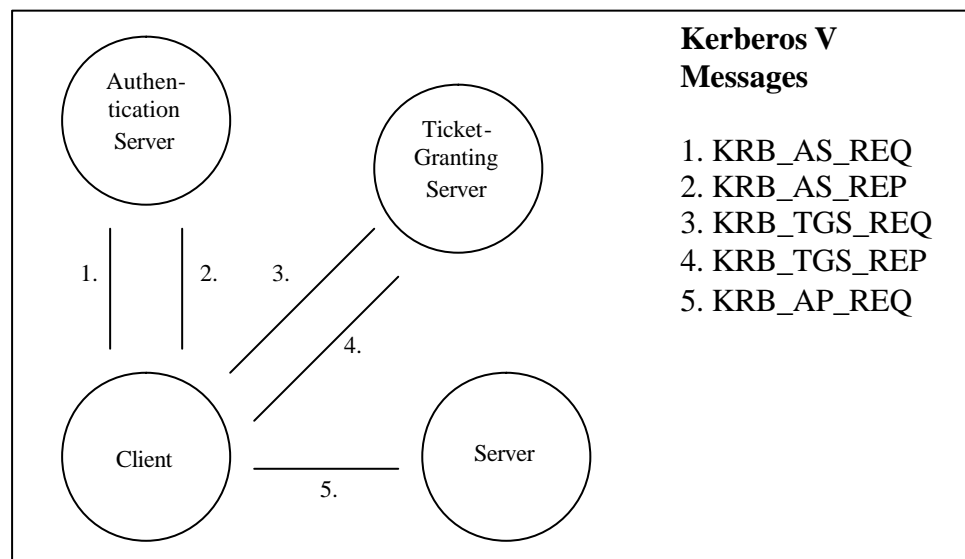
2 Background

This chapter will cover the technical details behind the Kerberos V and SMB protocols we discuss in this paper. First we will briefly cover Kerberos V, and next discuss essential parts of the Server Message Block (SMB) protocol that is used in the replay attack.

2.1 Kerberos V

Authentication in Kerberos V is based on symmetric cryptography and a trusted third-party that has access to all passwords. If a client wishes to authenticate himself to a network service, the typical process shown in Picture 1 includes the following steps:

1. The client requests for a Ticket-Granting Ticket (TGT) from an Authentication Server (AS)
2. The AS will reply with the TGT
3. The client will use the TGT to request for a ticket for the service he wishes to use from a Ticket-Granting Server (TGS)
4. The TGS will reply with a ticket for the service
5. The client will use the ticket and some other information to authenticate himself to the server



Picture 1. Kerberos authentication messages

For a detailed description of all the messages, we will refer the reader to [2]. To help understand the attacks described in later chapters, we will discuss some messages in more detail.

The first message in Picture 1, KRB_AS_REQ, is essentially of the form:

c, tgs, padata

Here *c* is the client's name, *tgs* the Ticket-Granting Server the client wishes to use and *padata* is pre-authentication data which the AS will use to validate the identity of the

client before sending a TGT back to the client. The *padata* field is optional, but is commonly used in a Windows 2000 environment because it makes offline password attacks harder to implement. Such attacks are still possible. The password attack we have implemented will attempt to exploit the characteristics of the pre-authentication data itself.

The last message, KRB_AP_REQ, includes a ticket for the service and something called an authenticator. The ticket includes the following information (note that $\{x\}K_y$ means x is encrypted with key K_y):

$$T_{c,s} = s, \{c, a, v, K_{c,s}\}K_s$$

Here s is the server's name, c the client's name, a the client's network address, v the validity time of the ticket, $K_{c,s}$ a session key between the client and the server, and K_s the server's key.

A new authenticator is created for each authentication attempt. Authenticators are of the following form:

$$A_{c,s} = \{c, t, seskey\}K_{c,s}$$

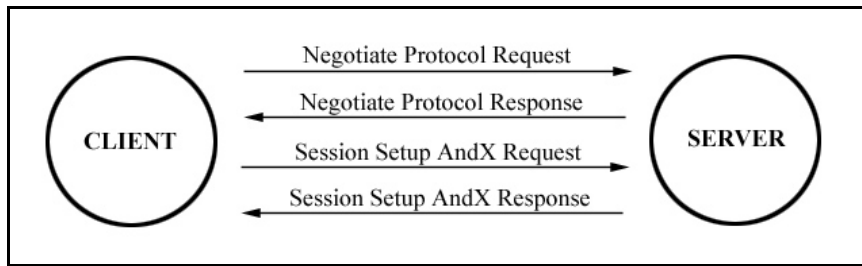
Here c is the client's name, t is a timestamp and *seskey* is an optional additional session key, all encrypted with the session key. Authenticators have two purposes. Firstly, they verify that the client knows the session key, because they include plaintext encrypted with it. Secondly, they include a timestamp, which should help to prevent replay attacks. It is also recommended that servers should cache used authenticators for the same reason.

The replay attack will exploit this final message sent to the server.

2.2 SMB

Server Message Block (SMB) is a protocol used for file and print services. It is a client-server, request-response protocol that enables servers to share file systems and other services. SMB primarily uses Kerberos V as an authentication mechanism in a Windows 2000 domain. It was created by IBM in the 1980's and has since evolved into many different dialects to support various needs. To those who are familiar with the acronym CIFS, it is the open version of SMB that some vendors are currently developing. The CIFS standard [3] can be used as a reference for the authentication process.

We will go through the session negotiation and authentication steps, which are important in understanding the replay attack we implemented. These steps are shown in Picture 2.



Picture 2. Messages transmitted in SMB session setup

We assume that both parties accept Kerberos as the authentication mechanism. There are normally four steps:

1. A client will send a Negotiate Protocol Request to the server, telling which dialects it supports.
2. The server responds with a Negotiate Protocol Response. It tells the client which dialect to use and other session information.
3. The client sends a Session Setup AndX Request to the server. In addition to other data, this message includes the KRB_AP_REQ described in the previous chapter.
4. The server responds with a Session Setup AndX Response indicating everything was acceptable

A weakness in this message exchange is exploited in the replay attack we implemented.

3 Threats and vulnerabilities

Kerberos V was created at MIT during project Athena to address several security issues concerning authentication. During that time, authentication often meant that plaintext passwords had to be sent over the network. Kerberos V imposes certain requirements for the environment. The Kerberos V standard [2] lists following assumptions:

- Denial of Service attacks are not solved with Kerberos V
- Secret keys must be kept secret
- Password guessing attacks are not solved by Kerberos V
- Clocks must be “loosely synchronized” to help prevent replay attacks
- Principal identifiers must not be recycled on a short-term basis

Kerberos V also implicitly relies on the servers being secure and software being non-malicious. However, the most interesting assumptions are the ones about password guessing and replay attacks. Both attacks are non-trivial but could be carried out over the local network. Password guessing attacks can be based on any text encrypted with the key derived from the victim’s password, and will result in exposure of the plaintext password. Replay attacks will usually result in the attacker assuming the victim’s identity without actually recovering the password. We will discuss both attacks in the next chapter.

4 Implemented Attacks

We focused our attacks on those vulnerabilities that could be exploited over the local network. In section 4.1 we discuss how an attacker might hijack a network connection allowing active monitoring and modification of the victim's network traffic. This is a precondition for the actual attacks against Kerberos V, which we discuss in sections 4.2 and 4.3.

4.1 Hijacking a Network Connection on a Switched Network

To hijack a network connection of our target machine we have to be able to direct the flow of network traffic from the target machine to our machine. The rest is accomplished by redirecting the packets in the kernel level.

The problem can be solved by the weaknesses of the Address Resolution Protocol (ARP) [4]. ARP is a stateless protocol which makes it legal to send ARP reply packets to the victim even if it has not send any ARP requests. This makes it possible for the attacker to send ARP reply packets continuously to the victim where the MAC address is forged to correspond to the one of the attacker's machine. Usually when you want to monitor the traffic originating from the victim, you need to spoof the gateway of the network.

Now we are able to listen to the network traffic from the victim machine. The packet redirection is accomplished with kernel tools such as *iptables* on Linux.

The objective of this attack was to fulfill the requirements issued by the attacks described in chapters 4.2 and 4.3.

Results of this attack are described in chapter 5.1.

4.2 Password Attack

The Windows 2000 implementation of Kerberos V protocol requires the use of the pre-authentication data in the KRB_AS_REQ message by default, which makes it harder to implement offline password attacks. If pre-authentication is not used, anyone can make a request for a TGT from the KDC (Key Distribution Center) and launch an offline password attack against it. The default implementation of pre-authentication data in Windows 2000 consists of an encrypted Kerberos timestamp created with a key derived from the user's password and a cryptographic checksum.

If an attacker is able to monitor the network traffic between the victim and the KDC server, a password attack becomes possible. This is based on the fact that before encryption the Kerberos timestamp is an ASCII-encoded string with the syntax "YYYYMMDDHHMMSSZ". This information makes it possible to find a valid password by running a dictionary or brute force attack against the encrypted timestamp. If the decrypted timestamp looks like a valid Kerberos timestamp we have found a candidate for the correct password. The correctness of the result can be verified by calculating the checksum. The exact descriptions of the cryptographic operations are described in [5].

A theoretical study on this matter has already been conducted and is available from [6].

The objectives of this attack were to prove to ourselves that the threat exists and to create necessary tools to make the exploitation of this vulnerability easier for us.

The results and the tools created are described in more detail in chapter 5.2.

4.3 Replay Attack

Replay attacks against Kerberos V are targeted on the final message transferred from the client to the server. This is the `KRB_AP_REQ` message also found in Picture 1. An attacker will attempt to capture this message and reuse its data to authenticate himself as the victim. If successful, the attacker will have full access to the same service the victim accessed. He will not, however, be able to recover the victim's actual password. This attack requires that traffic from the victim to the server is subverted to the attacker's network address. This can be achieved with a hijacking attack described in chapter 4.1.

We chose SMB as an example protocol that is susceptible to this attack. SMB authentication was discussed in more detail in chapter 2.3. There were two questions we wanted to answer with our research.

The first question was on the handling of the authenticators by the Windows 2000 Server. If a server does not cache used authenticators, replay attacks become much easier, as the attacker only has to passively monitor the network traffic from the victim. If a cache is used, the attacker has to actively prevent the server from seeing the security blob sent by the victim.

The other question was about the network addresses in tickets. The address included in a service ticket is optional according to [2], but is still highly recommended. This field would restrict the use of the ticket to pre-defined hosts, and make replay attacks more difficult, as the attacker would be forced to use the victim's IP address when replaying the credentials.

Results of such an attack are described in chapter 5.3. We also describe the tools we created to execute such an attack.

5 Analysis of Attacks

All of the attacks described in chapter 4 were carried out successfully. In this chapter we will present the results that we discovered when implementing the attacks. We also shortly describe the tools we created to repeat them. More technical descriptions of all the attacks can be found from [7], [8] and [9].

5.1 Hijacking a Network Connection on a Switched Network

The results we gained while implementing the hijacking attack show that the network traffic from the victim can be quite easily monitored and controlled by the attacker on

a switched network. The attack was conducted successfully which provided us the necessary tools to continue with the attacks against the Kerberos V protocol.

All the necessary tools to successfully complete these attacks are already implemented and freely available with source code from [10].

5.2 Password Attack

The results of this attack show that the pre-authentication scheme based on the symmetrically encrypted timestamp is indeed vulnerable to the dictionary and brute force attacks. It was quite trivial to gather pre-authentication data between the victim and the KDC server by passively monitoring the network traffic. Dictionary attacks were successfully launched against weak passwords. We can conclude that the feasibility of this attack depends mainly on the quality of the used passwords.

To make it easier to perform this attack we created two new tools. The first one [11] is a network sniffer which monitors the network traffic in promiscuous mode and collects pre-authentication data from the `KRB_AS_REQ` messages. The second program [12] performs a dictionary attack against the data collected with the first tool.

5.3 Replay Attack

We can conclude from the results of our research that replay attacks against SMB and Kerberos V on a Windows 2000 domain are quite feasible. An attacker will be able to use the victim's credentials to access file shares.

Our research shows that the Windows 2000 Server SP3 does indeed cache used authenticators. We attempted replaying used authenticators, but the server refused to accept them. This means that an attacker must use an active man-in-the-middle attack to listen in on the SMB session setup and prevent the server from ever seeing the credentials the victim sends. This way, when the attacker replays the security blob, the server has not seen the authenticator, and the attack succeeds.

Our research shows that the Windows 2000 Server SP3 acting as a file server either does not verify the address field or the Windows 2000 KDC does not include it in the tickets it gives out. This means that an attacker, once he has captured the victim's security blob, may reuse it from his own network address. This makes replay attacks simpler.

To execute such an attack, we created two tools. The first one [13] is a proxy that listens to connections on the attacker's machine, forwards session negotiations between the real server and the victim and captures the security blob inside the Session Setup AndX message. To replay this blob to the server, we patched [14] a program known as *smclient* from the Samba open source package [15] to use captured blobs when authenticating.

6 Protecting your Environment

In this chapter, we describe solutions to detect and prevent the attacks described in chapter 4.

6.1 Detecting and Preventing Network Connection Hijacking

Network connection hijacking can be done in many ways. Here we discuss the solutions against ARP spoofing.

There are generally two well known ways to detect ARP spoofing attempts – monitoring the local ARP cache and monitoring the network traffic on the wire.

ARP cache monitoring on a local machine can be accomplished with the *arp*-command. It is quite trivial to notice if the gateway's MAC address has changed (assuming the real MAC address of the gateway is known). This can be done automatically with a tool called *arpwatch* [16].

Network traffic monitoring can be implemented with certain Intrusion Detection Systems. The Open Source IDS called Snort [17] is able to do this in real time.

One of the best ways to protect machines against ARP spoofing attacks is to enforce static ARP entries on the local machines, especially the entry for the local gateway should be static.

6.2 Detecting and Preventing Password Attacks

This attack is accomplished by passively listening to the network traffic between the victim and the Kerberos KDC server. The only way to detect this is by monitoring the network for symptoms which might give us a hint that someone is running a sniffer on the network. More information can be found from [7].

This attack will become computationally infeasible if a strong password policy is implemented. The Windows 2000 implementation of Kerberos V supports also another pre-authentication method in addition to the password-based. This public key based scheme, called PKINIT [19], does not suffer from the weakness we are describing here. Another effective way to prevent this attack is to encrypt the network traffic, for example by using IPSEC.

6.3 Detecting and Preventing Replay Attacks

To detect a replay attack, one option would be to attempt detecting ARP spoofing altogether. This is described in more detail in chapter 6.1. If this is successful, the attack becomes infeasible.

The victim can also detect a possible attack if attempted connections seem to fail. When an attack is under way, the victim will see an error message stating that the service is not available. This is because the attacker will stop proxying traffic to the server after capturing the security blob. However, this is not an efficient solution,

since such errors are not rare in normal circumstances. Also, trusting users in such matters is probably not a good idea at all.

The detection of this attack is very difficult. More effort should be put into preventing it from happening. This is possible in a handful of ways. The most efficient way is to use some form of encryption on the IP layer. The use of IPSEC would be a sufficient protective action. However, using it to encrypt all client-to-server traffic is in many cases very difficult.

SMB signing, which is available on some implementations, can be used to prevent replay attacks. It is described in more detail in [3]. In short, when signing is enabled, packets will include a cryptographic MD5 checksum created with a session key to ensure their integrity. There is one significant pitfall. Servers usually support SMB signing, but don't require that clients always use it. If the victim is using SMB signing, the connection can still be attacked. The security blob is easily extracted, since no encryption is used. If the attacker is then allowed to connect to the server with the stolen credentials without signing, the attack will succeed.

The server must require SMB signing for all connections for the attack to fail. If this is the case, the attacker will not know the key to create the checksums, and therefore cannot create a connection.

If SMB connections have to be made in an unsafe network, it could be argued that other authentication methods such as NTLMv2 are in fact safer than Kerberos. Replay attacks on such challenge-response mechanisms are not possible, but dictionary attacks on weak passwords certainly are.

7 Conclusions

Our research shows that attacks against Kerberos V can be implemented with modest resources. We focused on attacks that can be carried out remotely requiring only physical access to the local network. An attacker is able to retrieve weak user passwords or abuse user credentials to access network resources without actually knowing the password.

The consequences are that Kerberos V alone does not guarantee secure authentication in an insecure network. Additional measures must be taken to ensure the vulnerabilities we have demonstrated cannot easily be exploited. The use of IPSEC is a solution, but domain-wide deployment is rarely an option because of additional overhead and technical limitations. The applications and protocols using Kerberos should also take countermeasures to ensure that replay attacks are not possible. SMB signing and other similar safeguards that verify the user's knowledge of actual session keys are warmly recommended.

Possible areas of future research include a more precise evaluation of the resources needed for a password attack against stronger passwords and the implementation of replay attacks against other protocols using Kerberos V. For example, LDAPv3 connections to Microsoft Active Directory are primarily authenticated with Kerberos V. A successful replay attack against an administrative LDAP connection will most probably allow an attacker to elevate his privileges by modifying user and

group objects stored in Active Directory. Our preliminary tests [20] show that replay attacks against LDAPv3 are possible.

8 References

- [1] S. M. Bellovin, M. Merrit. Limitations of the Kerberos Authentication System. <http://www.research.att.com/~smb/papers/kerblimit.usenix.pdf>. Referenced 26.3.2003.
- [2] J. Kohl, C. Neuman. The Kerberos Network Authentication Service (V5). Request for Comments 1510, September 1993.
- [3] Storage Networking Industry Association. CIFS Technical Reference 1.0. http://www.snia.org/tech_activities/CIFS/CIFS-TR-1p00_FINAL.pdf. Referenced 26.3.2003.
- [4] D. C. Plummer. An Ethernet Address Resolution Protocol. Request for Comments 826, November 1982.
- [5] M. Swift, J. Brezak. The Microsoft Windows 2000 RC4-HMAC Kerberos encryption type. Internet Draft draft-brezak-win2k-krb-rc4-hmac-04.txt, May 2002.
- [6] F. O'Dwyer. Feasibility of attacking Windows 2000 Kerberos Passwords. http://www.brd.ie/papers/w2kkrb/feasibility_of_w2k_kerberos_attack.htm. Referenced 26.3.2003.
- [7] K. Kasslin, A. Tikkanen. Hijacking a Network Connection on a Switched Network, March 2003.
- [8] K. Kasslin, A. Tikkanen. Password Attack on Kerberos V and Windows 2000, March 2003.
- [9] K. Kasslin, A. Tikkanen. Replay Attack on Kerberos V and SMB, March 2003.
- [10] D. Song. Toolset dsniff. <http://naughty.monkey.org/~dugsong/dsniff/>. Referenced 26.3.2003.
- [11] K. Kasslin, A. Tikkanen. sniff_krb5_asreq_packet.c.
- [12] K. Kasslin, A. Tikkanen. crack_krb5_preauth_data.c.
- [13] K. Kasslin, A. Tikkanen. smb_catchblob.c.
- [14] K. Kasslin, A. Tikkanen. Patch against samba-3.0alpha21.
- [15] Samba suite. <http://www.samba.org>. Referenced 26.3.2003.
- [16] LBNL's Network Research Group. <http://www-nrg.ee.lbl.gov/>.

Referenced 26.3.2003.

- [17] Snort.org. <http://www.snort.org/>. Referenced 26.3.2003.
- [18] R. Atkinson, S. Kent. Security Architecture for IP. Request for Comments 2401, November 1998.
- [19] B. Tung, C. Neuman, M. Hur, A. Medvinsky, S. Medvinsky, J. Wray, J. Trostle. Public Key Cryptography for Initial Authentication in Kerberos. Internet Draft draft-ietf-cat-kerberos-pk-init-16.txt, March 2002.
- [20] K. Kasslin, A. Tikkanen. Replay Attack against Kerberos V and LDAPv3, April 2003.