Stealing credentials for impersonation

Emmanuel Bouillon manu@veryopenid.net

October 29, 2010

Disclaimer

This expresses my own views and does not involve my previous, current and future employers. Presentation and code are provided for educational purpose only.

Outline

1. Introduction

2. Background

3. Pass the Ticket attack

4. Conclusion

Outline

1. Introduction
2. Background
3. Pass the Ticket attack
4. Conclusion

What is it about?

- User impersonation in Windows Active Directory domain
- Fully updated target
  - Windows Server 2008 R2 / Windows 7
  - No backward compatibility degrading security level (ex. Forest Functional Level)
- Practical implementation issues in realistic environments

Why should you care?

- Credentials theft for impersonation: key role in persistent intrusion
- Pervasive protocol in professional environments
    - Kerberos broadly accepted authentication protocol
    - Authentication protocol used by MS Active Directory services
- Common target moves from WS2003/XP to WS2008/W7
    - Influences credential theft for impersonation possibilities

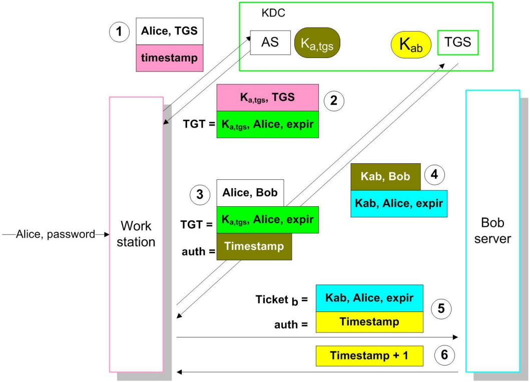
Outline

1. Introduction
2. Background
3. Pass the Ticket attack
4. Conclusion

Kerberos

Just what you need in mind

Text	cleartext
Text	ciphered with Alice's key
Text	ciphered with Bob's key
Text	ciphered with TGS's key
Text	ciphered with Alice & TGS session key
Text	ciphered with Alice & Bob's session key

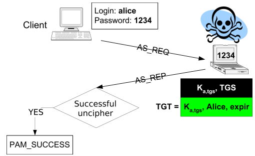


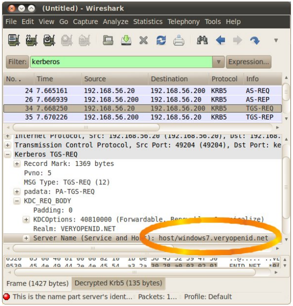
Old school ticket games - 1/2

Building stones for newer tricks

KDC spoofing

- Kerberos protocol precludes impersonation through KDC spoofing
- Lazy Kerberos based authentications are vulnerable
    - Ex. Badly configured PAM module
    - Easy to be vulnerable (Ex. Unix screen-savers)
- Windows implementation immune to basic KDC spoofing
    - Properly request a TS for host principal to validate TGT




(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: kerberos
Expression...

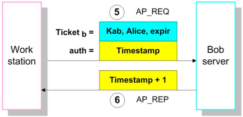
```
No. .  Time      Source          Destination     Protocol  Info
   24 7.665161   192.168.56.20   192.168.56.200  KRB5      AS-REQ
   26 7.666939   192.168.56.200  192.168.56.20   KRB5      AS-REP
   34 7.668250   192.168.56.20   192.168.56.200  KRB5      TGS-REQ
   35 7.670226   192.168.56.200  192.168.56.20   KRB5      TGS-REP

+ Internet Protocol, Src: 192.168.56.20 (192.168.56.20), Dst: 192.168.
+ Transmission Control Protocol, Src Port: 49204 (49204), Dst Port: ke
- Kerberos TGS-REQ
  + Record Mark: 1369 bytes
    Pvno: 5
    MSG Type: TGS-REQ (12)
  + padata: PA-TGS-REQ
  - KDC_REQ_BODY
      Padding: 0
    + KDCOptions: 40810000 (Forwardable, Ren[illegible]ize)
      Realm: VERYOPENID.NET
    + Server Name (Service and Ho[illegible]): host/windows7.veryopenid.net

0520  05 00 40 81 00 00 a2 10  1b 0e 56 45 52 59 4f 50   ..@..... ..VE
0530  45 4e 49 44 2e 4e 45 54  a3 2a 30 28 a0 03 02 01   ENID.NET .*0(

Frame (1427 bytes)   Decrypted Krb5 (135 bytes)
This is the name part server's ident...   Packets: 1...   Profile: Default
```

Old school ticket games - 2/2

Building stones for newer tricks

TS Replay

- TS and associated authenticator replay
- Means of mitigation
  - Time-based authenticators
  - Replay caches
    - Make passive network sniffing insufficient
    - Still vulnerable with active MitM attacks
  - Keyed cryptographic checksum can be included using the session key unknown by the attacker
  - Default configuration of MS Windows flavor



Newer implementation issues

KDC spoofing with PKINIT

- iSEC Partners - Attacking Kerberos Deployments - BlackHat US 2010 [1]
- Insider - with legitimate domain account - get the victim logged on under his account

Pass the Ticket

- Impersonate the victim during 10h after sniffing his/her authentication
- No valid credentials required for the bad guy
- Works locally **and** remotely[a]

[a] if Terminal Server enabled

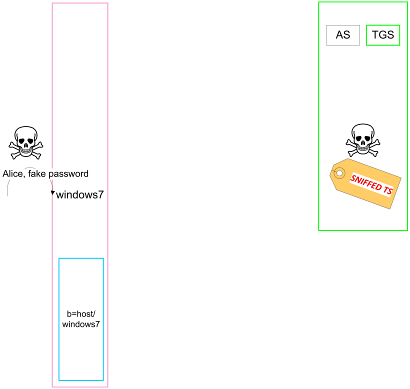
Outline

1. Introduction
2. Background
3. Pass the Ticket attack
4. Conclusion

Pass the ticket

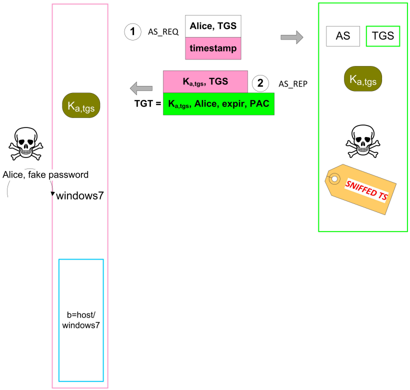
Principle

- Approach: sort of best effort, KDCspoof & Replay mix
- Sniff a valid TGS_REP for host/target_machine, save TS
- AS-REP: classical KDCspoofing
- TGS-REP: spoofed TGS-REP (based on AS-REP TGT) with a previously sniffed TS
    - Can't get sniffed TS session key
    - Can't generate a valid authenticator
- PAC = Privilege Attribute Certificate



Pass the ticket
Principle

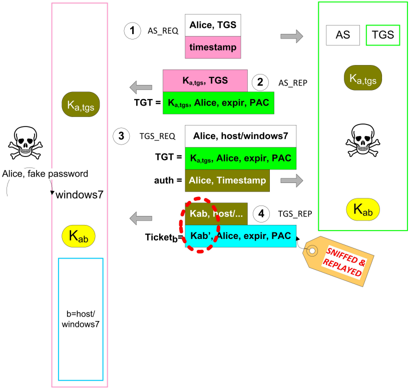
- Approach: sort of best effort, KDCspoof & Replay mix
- Sniff a valid TGS_REP for host/target_machine, save TS
- AS-REP: classical KDCspoofing
- TGS-REP: spoofed TGS-REP (based on AS-REP TGT) with a previously sniffed TS
  - Can't get sniffed TS session key
  - Can't generate a valid authenticator
- PAC = Privilege Attribute Certificate



Pass the ticket

Principle

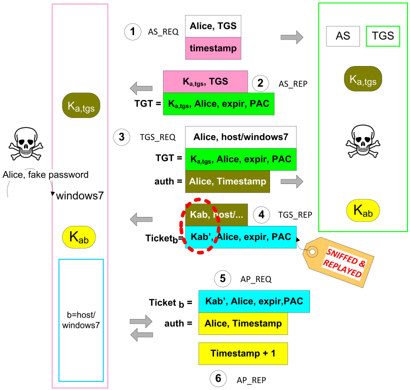
- Approach: sort of best effort, KDCspoof & Replay mix
- Sniff a valid TGS_REP for host/target_machine, save TS
- AS-REP: classical KDCspoofing
- TGS-REP: spoofed TGS-REP (based on AS-REP TGT) with a previously sniffed TS
    - Can't get sniffed TS session key
    - Can't generate a valid authenticator
- PAC = Privilege Attribute Certificate



Pass the ticket

Principle

- Approach: sort of best effort, KDCspoof & Replay mix
- Sniff a valid TGS_REP for host/target_machine, save TS
- AS-REP: classical KDCspoofing
- TGS-REP: spoofed TGS-REP (based on AS-REP TGT) with a previously sniffed TS
    - Can't get sniffed TS session key
    - Can't generate a valid authenticator
- PAC = Privilege Attribute Certificate



Alice, fake password

windows7

$K_{a,tgs}$

K_{ab}

b=host/ windows7

AS

TGS

$K_{a,tgs}$

K_{ab}

1 AS_REQ

Alice, TGS

timestamp

2 AS_REP

$K_{a,tgs}$, TGS

TGT = $K_{a,tgs}$, Alice, expir, PAC

3 TGS_REQ

Alice, host/windows7

TGT = $K_{a,tgs}$, Alice, expir, PAC

auth = Alice, Timestamp

4 TGS_REP

Kab, host/...

Ticket$_b$= Kab', Alice, expir, PAC

SNIFFED & REPLAYED

5 AP_REQ

Ticket $_b$ = Kab', Alice, expir,PAC

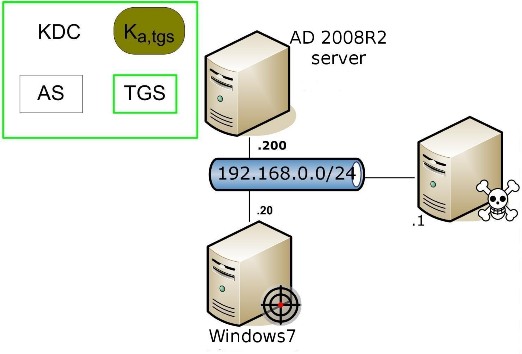
auth = Alice, Timestamp

Timestamp + 1

6 AP_REP

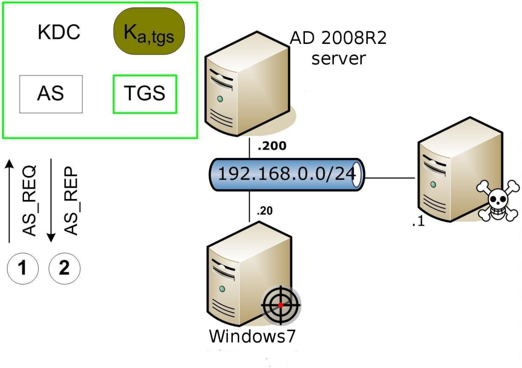
Attacks steps

1. Sniff a legitimate connection



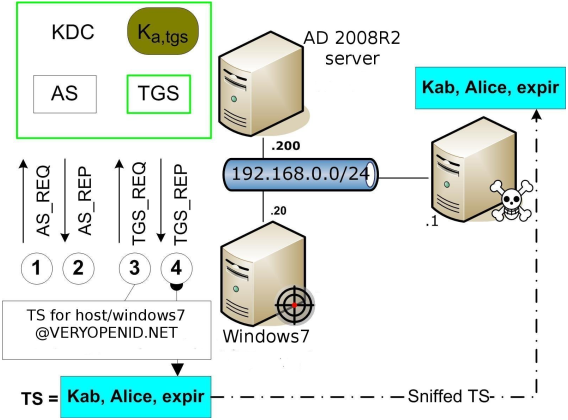
Attacks steps

1. Sniff a legitimate connection



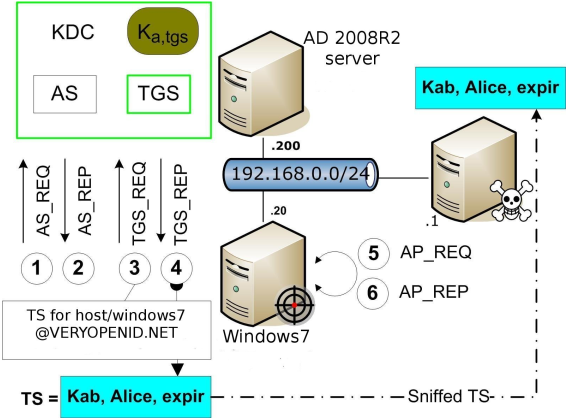
Attacks steps

1. Sniff a legitimate connection



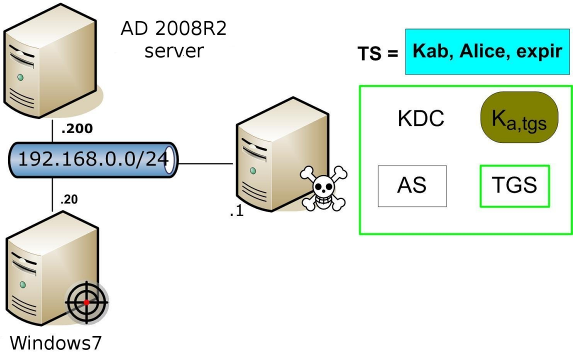
Attacks steps

1. Sniff a legitimate connection



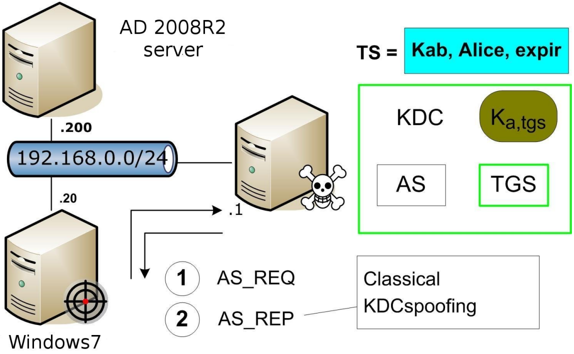
Attacks steps

2. Combine KDCspoof and replay



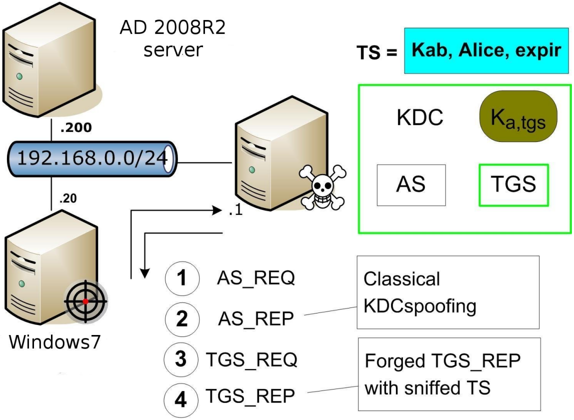
Attacks steps

2. Combine KDCspoof and replay



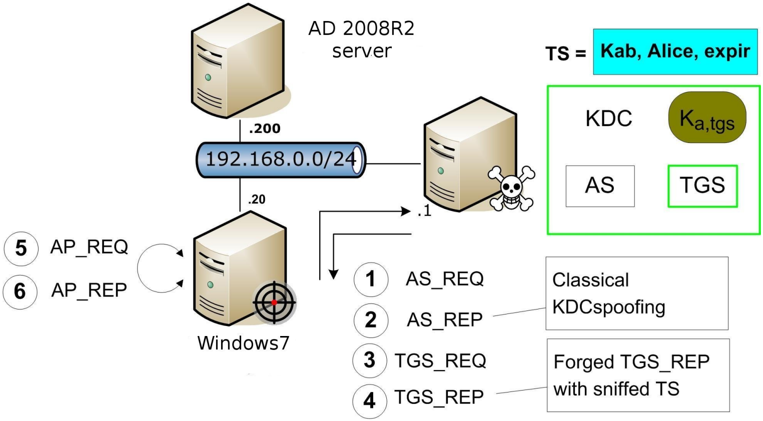
Attacks steps

2. Combine KDCspoof and replay



Attacks steps

2. Combine KDCspoof and replay



Demo

- Kerberos realm: VERYOPENID.NET
- Victim's account: Paul (real password: VeryG00dPwd!)

Implementation

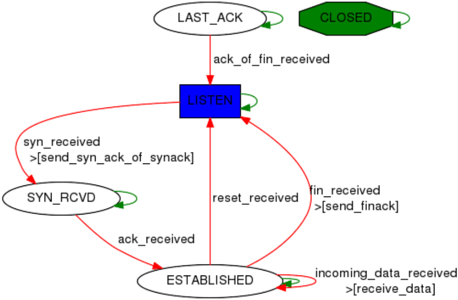
WS2003/XP − > WS2008/W7: Changes in defaults

- DES disabled
- Defaults to TCP not UDP
  - Already the case in realistic WS2003/XP environments
  - Possible to force UDP to TCP by sending a RESPONSE_TOO_BIG error message
  - Not possible to force TCP to UDP
- RC4-HMAC-MD5 changed to AES256-HMAC-SHA1 as default cryptosystem
- More robust against ARP cache poisoning?

Tools

Building blocks

- ASN.1: pyasn1
- MitM: ettercap
- Selective TCP connections spoofing: scapy automaton
  - Simulate just enough of a TCP/IP connection
- Changes in crypto: heimdal



Sample code at `http://code.google.com/p/krb5pyasn1`

What can be done

- Protect your layer 2
- Shorten service tickets lifetime
- Activate logs on Kerberos related events (both AD and Workstation)
    - And look at the logs
    - Think twice before implementing cross domain trust relationships
        - Probably better use claims based authentication and ADFS

Digression on disclosure time line

- June 2008: MSRC informed with documentation (XP)
  - "Normal Kerberos behavior"
- December 2008: PacSec Japan, demo to MS security expert

- October 2009: Thibault's security ad
  - "Windows7 was his idea" [2]

- December 2009: Windows 7 POC code
- June 2010: MSRC provided demo and source code and docs
  - "This is a valid issue, however we do not consider this a vulnerability". No CVE
- August 2010: Researchers (Venice University) release code [3] based on [4] harmless in most realistic situations

No sign that this is going to be solved quickly

Digression on disclosure time line

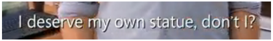
- June 2008: MSRC informed with documentation (XP)
    - "Normal Kerberos behavior"
- December 2008: PacSec Japan, demo to MS security expert

- October 2009: Thibault's security ad
    - "Windows7 was his idea" [2]
- December 2009: Windows 7 POC code
- June 2010: MSRC provided demo and source code and docs
    - "This is a valid issue, however we do not consider this a vulnerability". No CVE
- August 2010: Researchers (Venice University) release code [3] based on [4] harmless in most realistic situations

No sign that this is going to be solved quickly

Digression on disclosure time line

- June 2008: MSRC informed with documentation (XP)
  - "Normal Kerberos behavior"
- December 2008: PacSec Japan, demo to MS security expert
- October 2009: Thibault's security ad
  - "Windows7 was his idea" [2]

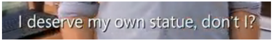


- December 2009: Windows 7 POC code
- June 2010: MSRC provided demo and source code and docs
  - "This is a valid issue, however we do not consider this a vulnerability". No CVE
- August 2010: Researchers (Venice University) release code [3] based on [4] harmless in most realistic situations

No sign that this is going to be solved quickly

Digression on disclosure time line

- June 2008: MSRC informed with documentation (XP)
    - "Normal Kerberos behavior"
- December 2008: PacSec Japan, demo to MS security expert
- October 2009: Thibault's security ad
    - "Windows7 was his idea" [2]

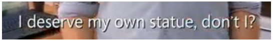


- December 2009: Windows 7 POC code
- June 2010: MSRC provided demo and source code and docs
    - "This is a valid issue, however we do not consider this a vulnerability". No CVE
- August 2010: Researchers (Venice University) release code [3] based on [4] harmless in most realistic situations

No sign that this is going to be solved quickly

Digression on disclosure time line

- June 2008: MSRC informed with documentation (XP)
  - "Normal Kerberos behavior"
- December 2008: PacSec Japan, demo to MS security expert
- October 2009: Thibault's security ad
  - "Windows7 was his idea" [2]



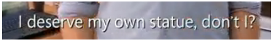
- December 2009: Windows 7 POC code
- June 2010: MSRC provided demo and source code and docs
  - "This is a valid issue, however we do not consider this a vulnerability". No CVE
- August 2010: Researchers (Venice University) release code [3] based on [4] harmless in most realistic situations

No sign that this is going to be solved quickly

Digression on disclosure time line

- June 2008: MSRC informed with documentation (XP)
  - "Normal Kerberos behavior"
- December 2008: PacSec Japan, demo to MS security expert

- October 2009: Thibault's security ad
  - "Windows7 was his idea" [2]

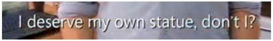


- December 2009: Windows 7 POC code
- June 2010: MSRC provided demo and source code and docs
  - "This is a valid issue, however we do not consider this a vulnerability". No CVE
- August 2010: Researchers (Venice University) release code [3] based on [4] harmless in most realistic situations

No sign that this is going to be solved quickly

Digression on disclosure time line

- June 2008: MSRC informed with documentation (XP)
  - ”Normal Kerberos behavior”
- December 2008: PacSec Japan, demo to MS security expert
- October 2009: Thibault’s security ad
  - ”Windows7 was his idea” [2]



- December 2009: Windows 7 POC code
- June 2010: MSRC provided demo and source code and docs
  - ”This is a valid issue, however we do not consider this a vulnerability”. No CVE
- August 2010: Researchers (Venice University) release code [3] based on [4] harmless in most realistic situations

No sign that this is going to be solved quickly

Outline

1 Introduction

2 Background

3 Pass the Ticket attack

4 Conclusion

Conclusion

- Since W2000, MS implements Kerberos as its default domain authentication protocol
- Greatly improves security compared to previous mechanism
- Default settings still more secure than the default Unixes implementation settings
  - If no clue on tuning a KDC, use AD
  - Please, Unixes guys, check if you can get a TS for your users!

BUT ...

- Implementation issue allows to bypass authentication
- Recent changes in default Kerberos implementation do not prevent Pass the Ticket attacks
- Not considered as a vulnerability by MSRC
- Better activate and monitor your logs

Conclusion

- Since W2000, MS implements Kerberos as its default domain authentication protocol
- Greatly improves security compared to previous mechanism
- Default settings still more secure than the default Unixes implementation settings
  - If no clue on tuning a KDC, use AD
  - Please, Unixes guys, check if you can get a TS for your users!

BUT ...

- Implementation issue allows to bypass authentication
- Recent changes in default Kerberos implementation do not prevent Pass the Ticket attacks
- Not considered as a vulnerability by MSRC
- Better activate and monitor your logs

Thanks for your attention

- Q & possibly A
- KRB5 pyasn1 module and sample code at `http://code.google.com/p/krb5pyasn1`

References I

- [1] iSEC Partners, Inc. - Attacking Kerberos Deployments - https://www.isecpartners.com/files/iSEC_BH2010_PKINIT_Preadvisory.pdf
- [2] Thibault's security ad - http://www.microsoft.com/showcase/en/us/details/8d75503e-068b-4f19-98dc-792094be9203
- [3] T. Malgherini and R. Focardi - http://secgroup.ext.dsi.unive.it/wp-content/uploads/2010/08/m0t-krb5-08-2010.pdf
- [4] E. Bouillon - Taming the beast - Assess Kerberos-protected networks - http://www.blackhat.com/html/bh-europe-09/bh-eu-09-archives.html