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
Kerberos
Just what you need in mind

1. Alice, TGS
   - timestamp
   - ciphered with Alice’s key
   - ciphered with TGS’s key
   - ciphered with Alice & TGS session key

2. TGT = $K_{a,tgs}, Alice, expir$
   - ciphered with Bob’s key
   - ciphered with Alice & TGS session key

3. Alice, Bob
   - TGT = $K_{a,tgs}, Alice, expir$
   - auth = $Timestamp$
   - cipered with Alice & Bob’s session key

4. Kab, Bob
   - Kab, Alice, expir

5. Ticket$_b$
   - auth = $Timestamp$
   - cipered with Alice & Bob’s session key

6. Bob server

E. Bouillon
Stealing credentials for impersonation
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

Stealing credentials for impersonation
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

E. Bouillon | Stealing credentials for impersonation
Newer implementation issues

KDC spoofing with PKINIT

- 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
Introduction
Background
Pass the Ticket attack
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

E. Bouillon
Stealing credentials for impersonation
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

E. Bouillon
Stealing credentials for impersonation
Introduction
Background
Pass the Ticket attack
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**

![Diagram of the Pass the Ticket attack](image-url)
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**
Attacks steps
1. Sniff a legitimate connection
Attacks steps
1. Sniff a legitimate connection

E. Bouillon
Stealing credentials for impersonation
Attacks steps

1. Sniff a legitimate connection
Attacks steps
1. Sniff a legitimate connection
**Attacks steps**

2. Combine KDCspoof and replay

```
AD 2008R2
192.168.0.0/24
Windows7
TS = Kab, Alice, expir
KDC
K_a,tgs
AS
TGS
```
Attacks steps
2. Combine KDCspoof and replay

![Diagram showing the steps of the attack]

- **Kab, Alice, expir**
- KDC
- K_{a,tgs}
- AS
- TGS
- Classical KDCspoofing

**Introduction**

**Background**

**Pass the Ticket attack**

**Conclusion**

E. Bouillon
Stealing credentials for impersonation
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


- 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?
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

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

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

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

No sign that this is going to be solved quickly
Digression on disclosure timeline

- 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

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

No sign that this is going to be solved quickly
Digression on disclosure timeline

- 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
Introduction

Background

Pass the Ticket attack

Conclusion

References I


