Windows Protocol Analysis: MSCHAP & Friends

Gros, Charles-Henri
Haley, David
Lisanke, Bob
Schaff, Clovis

Outline

- Overview of Windows Security Issues
- Various Protocols and Problems
- Introducing MSCHAP
- MSCHAP to MSCHAP2
- MSCHAP2 to PEAP
- Murφ Models
- Lessons Learned

An Encouraging Message

- Wed Mar 10, 6:55 PM ET

  SEATTLE (Reuters) - Microsoft Corp. (Nasdaq:MSFT - news) upgraded a recent security warning to "critical" after discovering new ways in which an attacker could run malicious software on a vulnerable computer, the world's largest software maker said on Wednesday.

  The software flaw, which affects the two latest versions of Microsoft's Outlook e-mail, calendar and contacts program, were initially rated as "important" in Microsoft's monthly security bulletin issued on Tuesday.

A Horde of Protocols

- Transport Layers
  - NetBIOS, NetBEUI, TCP/IP…
- Protocols on top
  - SMB, RPC, NetMeeting…
- Many dialects of protocols
  - SMB: PCNP1.0, LanMan 1.0/2.0, NT LM 0.12, CIFS…

Lots of Protocols = Lots of Problems

- Backwards compatibility between all various dialects
- More implementations: more potential for human error (incorrect code…)
- Most protocol weaknesses seem unrelated to the protocol itself

Implementation Flaws

- Old friends like Buffer Overflows
- Holes in client-side code (ActiveX…)
- Poor crypto implementation might be easier to crack
- Programmer Laziness/Carelessness

Troubleshooting "Humanware"

- Windows empowers the user, less restrictive environment
- Easy for the unwary user to execute unwanted code (email virus)
- Convenience vs. Security (automatic parsing of HTML email, etc.)
- Uneducated user = highly vulnerable

The Password Paradigm

- Completely and utterly depends on secrecy and strength of password
- Many ways to fool uneducated user into giving away password (impersonating administrators, etc.)
- Reused password = less secure

Windows Protocols

- Hard to find current specifications
- Hard to tell off-hand why some services are running, others aren't
- Many are activated for unclear reasons (e.g. SQL server)
- To understand requires a competence which most end-users lack

Where did all the specs go? Long time passing…

- There seem to be no formal specs for CIFS (protocol for Windows file-sharing)
  - "Without a current and authoritative protocol specification, there is no external reference against which to measure the 'correctness' of an implementation, and no way to hold anyone accountable. Since Microsoft is the market leader […] the behavior of their clients and servers is the standard against which all other implementations are measured."
    Christopher Hertel, http://www.ubiqx.org/cifs/SMB.html

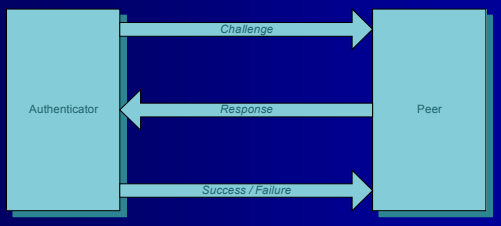
Chosen Area: Point to Point Authentication

- Windows supports:
  - Password Authentication Protocol
  - CHAP: Challenge-Handshake Authentication Protocol
  - MSCHAP: MS extensions to CHAP
  - MSCHAP2: Fixes to MSCHAP
  - Others (EAP, PEAP…)
- PAP: passwords transmitted in plaintext
- Acceptable before when networks were very small
- (MS)CHAP's major improvement: passwords no longer transmitted in plain text!
- Sounds good…

But…

CHAP does not specify which encryption algorithm to use. MSCHAP on the other hand, does.

CHAP Protocol



Events & Background

- August 1996
  - RFC 1334: CHAP
- Oct 1998
  - RFC 2433: MSCHAP1
- Jan 2000
  - RFC 2759: MSCHAP2
- Nov 2001
  - 1.4 Update to Win98 Dial-Up-Networking, implements MSCHAP2
- Oct 2003: PEAP Internet Draft
  - Protected Extensible Authentication Protocol. Combines TLS and MSCHAP2.

Cryptanalysis of Microsoft's Point to Point Tunneling Protocol (PPTP)

Schneier & Mudge (98)

- For Virtual Private Network, connection over TCP/IP link
- Microsoft's implementation breaks down:
  - Authentication level = MS-CHAP
  - Encryption = RC4
- Point to Point Tunneling Protocol: data channel encapsulated in PPP packets;
  - no protocol specification for security
- MS-PPTP: server under WinNT
  - auth. options: clear password, or hashed, or challenge-response

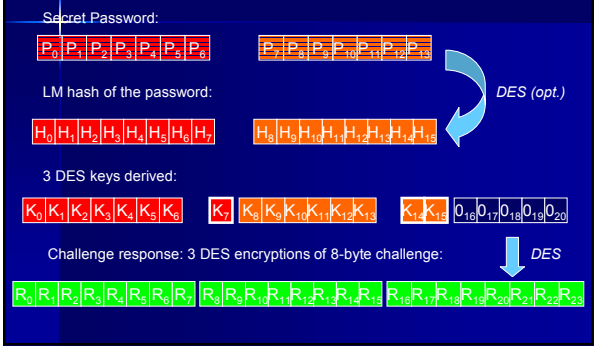
MS-PPTP Cryptanalysis Part 2 – LanMan Hash

- Windows NT hash functions:
  - LanManager hash based on DES; Win NT hash based on MD4
- LM's hash is "home-made" and weak:
  - truncates password to 14-char string;
  - converts lowercase to uppercase;
  - splits 14-byte in two 7-byte halves, giving two DES keys
  - with keys, encr. magic "KGS!@#$%" -> 2 8-byte strings
  - concatenate those string : 16-byte hash value
- WinNT hash: 16-byte hash with MD4, no salt either

MS-PPTP Cryptanalysis Part 3 – MS-CHAP Challenge

- MS-CHAP Challenge-Response step:
  - Authenticator *Challenge*:
    - 8-byte random value
  - Client side: for both LM and NT hash function…
    1. computes 16-byte hash value
    2. Zero-Pad to get to 21-byte value -> 3 7-byte DES keys
    3. encrypt challenge with each DES key
    4. concatenate those 3 8-byte values -> 24-byte response
  - Client *Response*:
    - send back both values, with a flag

MS-PPTP cryptanalysis Part 4 – Challenge view



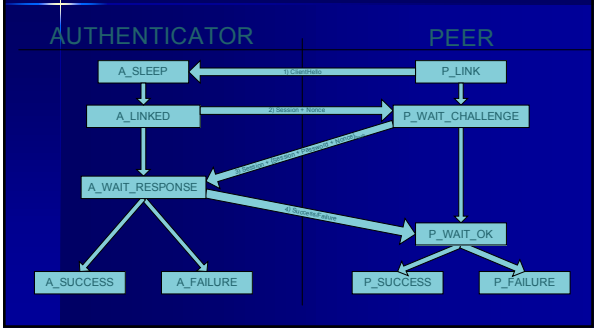
MS-PPTP cryptanalysis Part 5 – Attack on MS-CHAP

- Cryptanalysis of MS-CHAP:
  - Dictionary attack [LOpht proved it is efficient]
  - Offline: pre-computed DES encryption of each likely values of P0…P6 and P7…P13
  - Given $R_0…R_7$ $R_8…R_{15}$ $R_{16}…R_{23}$ seen on link:
    1. Retrieve K_{14} and K_{15} : average 2^{15} DES ops.
    2. for N_2 likely values of $P_7…P_{13}$: (DES encr. known) K_{14} and K_{15} retrieved : $N_2/2^{16}$ DES trials max
    3. for N_1 likely values of $P_0…P_6$: K_7 retrieved : $N_1/2^8$ DES trials max
- Cryptanalysis of MS-PPE: secret key also based on password

The 'LOpht' Crack on the LanMan Password Hash

- Creator: Mudge, Schneier's co-author of the article
- April 97, *Electronic Engineering Times*: Explanation of Mudge's motivations; Nash, MS 'director of marketing for Windows NT Server', answers back.
  - Mudge would like to have MS policy on security changed;
  - Nash claims enough internal beta-testing
- July 98, *Windows & .NET magazine*: 'NT Server Security Checklist' excerpts…
  - Enforce strong password policy
  - Use password crackers:
    - "The latest version of L0phtCrack is Microsoft's worst nightmare and every NT administrator's new best friend."

Murφ Modeling of CHAP (RFC 1994)

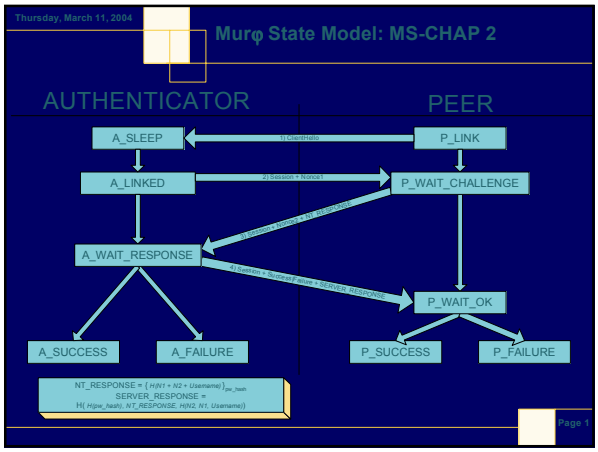


(MS)CHAP1 Problems

- CHAP and MSCHAP both suffer from man-in-the-middle (no server authentication). Murφ verified this.
- MSCHAP1: Failure_PasswordExpired forces bad LanMan hash to be sent

Thus Came MSCHAP2

- MSCHAP2 addresses two points:
  - Cryptography: uses SHA-1, MD4
  - Man-in-the-middle partially solved: server authentication through client challenge
- Client sends its own challenge along with its response
- In success message server sends monster-hash back



MSCHAP2

- To be able to generate response hash, one needs to have the plain-text or 1-step hashed password available.
- According to Murφ however there is still a man-in-the-middle attack
- Solution: send server's name in the hash
- MSCHAP2 still depends on password integrity!
- Microsoft decided to keep backwards compatibility with MSCHAP1 – so the attacker can convince both the client and server to negotiate that instead!

Modeling Procedure

- Modeled CHAP – discovered basic attack (MitM)
- Modeled MSCHAP1 – verified MitM, and that intruder could convince client to send LanMan hash
- Modeled MSCHAP2 – but ran into a wall

Modeling Difficulties

- Schneier article "polluted" first attempt.
  - We knew what we wanted to show, so we designed the model to show it!
  - Left out many possible intruder moves
  - Model "felt bad" and was obviously incomplete

→ Redesigned model to have a much more robust intruder.

→ This confirmed MitM for MSCHAP2, which did not appear with weaker model

Conclusions

- Hard to sort through morass of informal specifications
- MSCHAP2 seems to fix MSCHAP1 problems, but allows for version rollback attacks
- Murφ seems adequate for this protocol
- However, the found attacks are obvious enough after having formalized the RFCs

Conclusions, cont'd

- MSCHAPv2: better crypto, but still only as secure as password
- Backwards compatibility removes much of the point of an upgrade – both for MSCHAPv1 (LanMan hash) and MSCHAPv2 (compatibility with v1)
- MSCHAPv1 mistake (poor hash) should have been avoided
  - Improper, insufficient cryptanalysis
- Big problem with MSCHAPv1 is not the fault of the protocol itself
- MSCHAPv2: more robust crypto, but protocol is still flawed

References

- RFCs
  - http://www.zvon.org/tmRFC/RFC2759/Output/index.html
  - http://www.zvon.org/tmRFC/RFC2433/Output/index.html
  - http://www.zvon.org/tmRFC/RFC1994/Output/index.html
- Schneier papers:
  - http://www.schneier.com/paper-pptp.html
  - http://www.schneier.com/paper-pptpv2.html

References, cont'd

- MS Knowledge Base
  - Articles 297816, 285189, 297840, 297818
- MSDN:
  - http://msdn.microsoft.com/library/en-us/wceeap/html/cxconextensibleauthenticationprotocol.asp
- SMB/CIFS:
  - *What is SMB?*, Richard Sharpe, 2002, http://samba.org/cifs/docs/what-is-smb.html
  - *Implementing CIFS*, Christopher R. Hertel, 2003, http://www.ubiqx.org/cifs/