CMSC 426
Principles of Computer Security

Lecture 16

Linux and Windows Authentication

Last Class We Covered

- Authentication
- Password Hashing
- Password Cracking
  - Brute-force attacks
  - Dictionary attacks
  - Rainbow tables
- Salted passwords

Any Questions from Last Time?

Recap: Different Password Attacks

- Brute-force
  - Try all possible passwords on the system (no hashes needed)
  - BIG time investment, zero space cost
- Dictionary attack (data structure)
  - Create a dictionary of possible passwords, where the key is the hash, and the value is the password
  - BIG upfront time investment, BIG BIG space cost, small attack time
- Rainbow tables
  - Use a different method of storing the possible passwords and hashes
  - BIG upfront time investment, BIG space cost, medium attack time

Dictionary attack ("traditional")
Use a list of possible passwords; can be combined/used with other attacks

Recap: Different Password Attacks

- Brute-force
  - Try all possible passwords on the system (no hashes needed)
  - BIG time investment, zero space cost
- Dictionary attack ("traditional" definition)
  - Use a list of possible passwords; can be used with other attacks
- Dictionary attack (data structure definition)
  - Create a dictionary of possible passwords, where the key is the hash, and the value is the password
  - Upfront time investment, BIG BIG space cost, small attack time

Today's Topics

- Linux authentication

- Windows authentication
  - Standalone system authentication
  - Domain authentication

- ~~Kerberos protocol~~
  - Next time

Linux Authentication

The /etc/passwd File

- Text file that stores information about user accounts
  - Readable by any user
  - On legacy UNIX systems, stored passwords hashed by the UNIX **`crypt()`** algorithm
  - Now a bit of a misnomer on modern Linux operating system

- For each user, lists username, user ID (uid), group ID (gid), comments, home directory, preferred shell

```
rj:x:1000:1000:RJ,,,:/home/rj:/bin/bash
```

UNIX `crypt()` Algorithm

- UNIX passwords were originally limited to 8 characters

- **`crypt()`** was a legacy function used to “hash” passwords
  - Password used was a 56-bit key
  - 12-bit salt
  - Used a modified version of DES to make it one-way
  - Encrypted a 64-bit block of 0s, 25 rounds

The /etc/shadow File

- Text file that stores password information, only readable by root
- For each user, lists username, hashed password, info about password change/expiration policy
- Password field actually contains 3 separate parts separated by **$**
  - ID of hashing algorithm, salt, password hash

```
rj:$6$VEpaqG7Z$0hdWp6bdmrrgNX/44msduyOkd5W8fhIe
a1cZWcvrIv0rVNw2PWxPugoKmRNeqrptbR5tGjOo10UFVZl
pQlnIk1:17416:0:99999:7:::
```

Windows Authentication

Local Security Authority (LSA)

- Windows subsystem responsible for managing authentication and local security policy

- Local security policy determines:
  - Which users can access the system and in what way (*e.g.*, interactively, over the network, or as a service)
  - Which users have which permissions on the system
  - What forms of auditing are being performed

Security Accounts Manager (SAM)

- Database on standalone Windows systems that stores users' password hashes
- Two password hashing algorithms have been used
  - Lan Manager hash (LanMan, LM)
  - NT hash (NTLM)
- On most modern Windows versions, the SAM file is additionally encrypted to prevent offline password cracking
  - As of Windows 10, full disk encryption is preferred

Lan Manager Hash (LanMan, LM)

- In legacy versions of Windows, passwords were 14 characters max and not case sensitive

- LM hash algorithm:
  - Pad password to 14 characters
  - Convert to upper case
  - Split in half, use each half as a 56-bit DES key
  - DES encrypt the string "**KGS!@#$%**" with both keys
  - Concatenate the two encrypted strings

Lan Manager Hash Security

- Trivial to brute force
  - Just need to crack each of the two 7-byte halves
  - Exponentially easier to brute-force two 7-character strings than a single 14-character string
  - Also, since passwords are converted to uppercase, it's even easier to crack
- Default prior to Windows NT (released 2003)
- Disabled since Windows Vista
  - Means it was usable until 2006!

NT Hash (NTLM)

- Hash used by modern Windows systems

- NT Hash Algorithm:
  - Encode password in UTF-16 little-endian
  - Take the MD4 hash of the encoded password

- Other info:
  - Passwords can be longer than 14 characters
  - Passwords are still not salted

Windows Domains

- A group of computers connected over a network and managed by a central computer called a domain controller
- Password hashes stored in Active Directory database
- Three protocols have been used for authentication between a client and a server on a domain
  - NTLMv1 Protocol
  - NTLMv2 Protocol
  - Kerberos

NTLMv1 Authentication Protocol

- Server issues a random 8-byte challenge to the client
- Client computes both the LM and NT hashes of the password
- Each 16-byte hash is padded to 21 bytes using 5 null bytes
- Both 21-byte values are separated into three 7-byte (56-bit) blocks
- Each block is DES encrypted using the challenge as a key, then all are appended together into a 48-byte response
- The server computes this as well and validates the response

- Insecure due to use of LM hash and DES

NTLMv2 Authentication Protocol

- Server issues a random 8-byte challenge to the client
- Client sends two responses containing information such as a random 8-byte value, the current time, the NT hash of the password, the user name, and the name of the domain
- Server validates responses

```
SC = 8-byte server challenge, random
CC = 8-byte client challenge, random
CC* = (X, time, CC2, domain name)
v2-Hash = HMAC-MD5(NT-Hash, user name, domain name)
LMv2 = HMAC-MD5(v2-Hash, SC, CC)
NTv2 = HMAC-MD5(v2-Hash, SC, CC*)
response = LMv2 | CC | NTv2 | CC*
```

Kerberos Protocol

Kerberos Protocol

- Leading standard protocol for remote authentication
  - Used by many OSes, not just Windows
  - Will mostly be talking about it in the context of Windows domains
- Manages client-server interactions using a Key Distribution Center (KDC)
- Key Distribution Center provides two services:
  - Authentication Service (AS)
  - Ticket-Granting Service (TGS)