CSE 543 - Computer Security

Lecture 7 - Authentication
September 18, 2007
URL: http://www.cse.psu.edu/~tjaeger/cse543-f07/
What is Authentication?

• Short answer: establishes identity
  – Answers the question: To whom am I speaking?

• Long answer: evaluates the authenticity of identity proving credentials
  – Credential – is proof of identity
  – Evaluation – process that assesses the correctness of the association between credential and claimed identity
    • for some purpose
    • under some policy
Why authentication?

• Well, we live in a world of rights, permissions, and duties?
  – Authentication establishes our identity so that we can obtain the set of rights
  – E.g., we establish our identity with Tiffany’s by providing a valid credit card which gives us rights to purchase goods ~ physical authentication system

• Q: How does this relate to security?
Why authentication (cont.)?

• Same in online world, just different constraints
  – Vendor/customer are not physically co-located, so we must find other ways of providing identity
    • e.g., by providing credit card number ~ electronic authentication system
  – Risks (for customer and vendor) are different
    • Q: How so?

• Computer security is crucially dependent on the proper design, management, and application of authentication systems.
What is Identity?

• That which gives you access … which is largely determined by context
  – We all have lots of identities
  – Pseudo-identities

• Really, determined by who is evaluating credential
  – Driver’s License, Passport, SSN prove …
  – Credit cards prove …
  – Signature proves …
  – Password proves …
  – Voice proves …

• Exercise: Give an example of bad mapping between identity and the purpose for which it was used.
Credentials

• ... are evidence used to prove identity
• Credentials can be
  – Something I am
  – Something I have
  – Something I know
Something you know …

- Passport number, mothers maiden name, last 4 digits of your social security, credit card number

- Passwords and pass-phrases
  - Note: passwords are generally pretty weak
    - University of Michigan: 5% of passwords were goblue
    - Passwords used in more than one place
  - Not just because bad ones selected: If you can remember it, then a computer can guess it
    - Computers can often guess very quickly
    - Easy to mount offline attacks
    - Easy countermeasures for online attacks
Something your have ...

- Tokens (transponders, …)
  - Speedpass, EZ-pass
- Smartcards

- Digital Certificates (used by Websites to authenticate themselves to customers)
  - More on this later …
Something your are …

• Biometrics measure some physical characteristic
  – Fingerprint, face recognition, retina scanners, voice, signature, DNA
  – Can be extremely accurate and fast
  – Active biometrics authenticate
  – Passive biometrics recognize

• What is the fundamental problem?
  – Revocation – lost fingerprint?
  – Great for physical security, generally not feasible for on-line systems
Web Authentication

• Authentication is a bi-directional process
  – Client
  – Server
  – Mutual authentication

• Several standard authentication tools
  – Basic (client)
  – Digest (client)
  – Secure Socket Layer (server, mutual)
  – Cookies (indirect, persistent)

• Q: Are cookies good credentials?
How Basic Authentication Works …

CLIENT

GET /protected/index.html HTTP/1.0

HTTP/1.0 401 Unauthorized
WWW-Authenticate: Basic realm=“Private”

CLIENT

GET /protected/index.html HTTP/1.0
Authorization: Basic JA87JKAs3NbBDs
Setting up Basic auth in Apache

• File in directory to protect (.htaccess)

```
AuthType Basic
AuthName Trent's directions (User ID=jaeger)"
AuthUserFile /usr/jaeger/www-etc/.htpw1
AuthGroupFile /dev/null
require valid-user
```

• In /usr/jaeger/www-etc/.htpw1

  jaeger:17FwWEqjyzmNo

generated using htpasswd program

• Can use different .htaccess files for different directories
Basic Authentication Problems

- Passwords easy to intercept
- Passwords easy to guess
  - Just base-64 encoded
- Passwords easy to share
- No server authentication
  - Easy to fool client into sending password to malicious server
- One intercepted password gives eavesdropper access to many documents
Digest Authentication

GET /protected/index.html HTTP/1.1

HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
realm="Private" nonce="98bdc1f9f017.."

GET /protected/index.html HTTP/1.1
Authorization: Digest
username="lstein" realm="Private"
nonce="98bdc1f9f017.." response="5ccc069c4.."
Challenge and Response

- **Challenge ("nonce"): any changing string**
  - e.g. $\text{MD5(IP address:timestamp:server secret)}$

- **Response: challenge hashed with user’s name & password**
  - $\text{MD5(MD5(name:realm:password):nonce:MD5(request))}$

- **Server-specific implementation options**
  - One-time nonces
  - Time-stamped nonces
  - Method authentication digests
Advantages of Digest over Basic

- Cleartext password never transmitted across network
- Cleartext password never stored on server
- Replay attacks difficult
- Intercepted response only valid for a single URL
- Shared disadvantages
  - Vulnerable to man-in-the-middle attacks
  - Document itself can be sniffed
Kerberos Issues

• Bellovin and Merritt

• Weaknesses
  • System Issues
  • Replay
  • Passwords
  • Cryptanalysis

• As is typical, cryptanalysis seems to be the least feasible
Client Weaknesses

- Clients are not secure computing environments
  - But, session keys are stored there

- Authentication Key must be plaintext at some point
  - Host security may not protect key from attackers

- Session Key is cached for use
  - Host security may not protect

- Where is are the keys cached and stored?
  - Written to swap?
  - Remote file server

- All keys on client are vulnerable!
Replay

• Kerberos uses timestamp in its authenticator
  • Live for 5 minutes

• Claim: Susceptible to replay
  • Capture a ticket and authenticator for a legitimate session
    • Authenticator: \( \{c, \text{IP}_C, \text{timestamp}\}K_{CS} \)
  • Replay from \( \text{IP}_X \)
    • Ticket + Authenticator to Service
    • Resend messages of \( c \) from \( \text{IP}_X \)

• Verify for yourself that \( S \) would respond

• Also: Mess with time services to replay stale authenticators
Password Attacks

- Use of passwords in Kerberos is susceptible to **offline cracking**

- Process:
  - User enters password for Kerberized client
  - Request (w/o password) forwarded to KDC
  - Response is encrypted in key derived from user’s passwd
  - Client generates key from password for decryption

- Attack: If you know what the message should say, you can guess and test passwords

- We ran this and recovered 35% of CSE passwds

- Can also spoof logins to recover passwds
What are you gonna do?

- Secure storage
  - How would this work? Compare to *Trusted Computing*

- Protocol implementation changes
  - Challenge-response (prevent replay)
  - Change login protocol (prevent guessing)

- Others
  - Standardized message encoding (remove ambiguity)
  - Multi-session keys to compute session key (reduce exposure)
  - Standardize encryption components (reduce implementation errors, e.g., poor randomness)
Result of Bellovin-Merritt Warnings

- Community Acceptance
  - Their points were accepted

- Very Little Impact on Kerberos Function
  - No tangible change in the protocol
  - Even to V5

- Why not?
Cryptanalysis and Protocol Analysis

- Cryptographic Algorithms
  - Complex mathematical concepts
  - May be flawed
  - What approaches are used to prove correct/find flaws?

- Cryptographic Protocols
  - Complex composition of algorithms and messages
  - May be flawed
  - What approaches are used to prove correct/find flaws?
Cryptanalysis of RSA

• Survey by Dan Boneh
  – Real heavy math

• Results
  – Fascinating attacks have been developed
  – None devastating to RSA

• Cautions
  – Improper use
  – Secure implementation is non-trivial
Cryptanalysis of RSA

• Security Premise
  – Factoring Large Integers is Hard
  – \( N = pq \); \( N \) is known, can we find \( p, q \)

• Some Known (to cryptanalyst)
  – If \( (p-1) \) is product of prime factors less than \( B \)
  – \( N \) can be factored in time less than \( B^3 \)

• Best Known Approach: General Number Field Sieve
  – Significant early application by Arjen Lenstra
  – Current Status (May 2005)
    • German Federal Agency for Information Technology Security
    • Factor 663-bit number
    • Took “several months” using 80 AMD Apteron CPUs
Misuse of RSA

• Common Modulus Misuse
  – Use the same $N$ for all users
  – Since all have a private key for same $N$
    • Anyone can factor

• Blinding Misuse
  – Suppose adversary wants you to
    • Sign an arbitrary message $M$
  – You don’t sign
  – Adversary generates innocent $M’$
    • Where $M’ = r^e M \mod N$
    • Adversary can generate signature of $M$ from $M$’s signature

• Only use RSA (or any algorithm) in standard ways
RSA Exponent Problems

• Small Private Exponent
  – Speeds decryption time

• However, Known Attacks Exist on Small Private Keys
  – Due to Mike Wiener, can recover private key
  – Result: If N is 1024 bits, d of private key must be at least 256 bits long
  – Some workarounds are known (e.g., based on Chinese Remainder Theorem), but not proven secure

• Small Public Exponent
  – Speed signature verification time
  – Smallest possible value is 3, but recommend $2^{16} + 1$
  – Can recover M encrypted with multiple, small public keys
  – Can recover private key from small public + bits of private
Timing Attacks

- Use the timing behavior of system to extract secret
- Suppose a smartcard stores your private key
  - By precisely measuring the time it takes to perform private key ops, we can recover the key
  - Due to Kocher
  - At most $2^n$ operations required, where $n$ is the number of bits in the key
- Attack summary
  - Adversary asks smartcard to generate signatures on several messages
  - Recover one bit at a time starting with least significant
  - Compare times to those measured offline
- Solution: blinding
Power Analysis Attacks

• Also, Discovered by Kocher
  – Power usage is higher than normal in these computations
  – Measure the timing of high power consumption

• Simple Power Analysis
  – Direct interpretation of power measurements
  – Reveals instructions executions
  – Some crypto ops may be sensitive to data, e.g., DES S-boxes

• Differential Power Analysis
  – Statistical analysis of power data correlations

• Solution: Gotta change the code
Power and Timing

- What is the threat model in power/timing attacks?
- How does this conflict with the trust model?
- What is the vulnerability?