CS 5410 - Computer and Network Security: Authentication

Professor Patrick Traynor
Fall 2017
Announcements

• Assignment #2 is due on 9/18.
  • Submit via Canvas.
• Project Ideas are due on Wednesday
  • Also submitted via Canvas
• Install “John the Ripper” onto your laptops for Friday.
What is research?

• Which activities are research?
  • Designing a new protocol?
  • Building an implementation of a protocol?
  • Measuring the cost of the protocol?
  • Formally evaluating the correctness of a protocol?
  • Developing methods of implementing/evaluating a protocol?
What is not research?

• Arguing the quality of a protocol?
• Arguing the appropriateness of a protocol?
• Surveying a field?
• Illustrating a limitation of a common practice or system?
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 (what constitutes a good cred.?)
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
    ‣ How is this different from “Authorization”?

• 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 …

- 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 “only you” know?

- Password reuse?
- What about security questions?
  - Mother’s Maiden Name
  - Social Security Number
  - First pet
- Does anything make a good security/secret question?
Password Attacks

- Use of passwords in, for example, 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 passwords

- We will try some password cracking soon...
A petard ...

- The rule of seven plus or minus two.
  - George Miller observed in 1956 that most humans can remember about 5-9 things more or less at once.
  - Thus is a kind of maximal entropy that one can hold in your head.
  - This limits the complexity of the passwords you can securely use, i.e., not write on a sheet of paper.
  - A perfectly random 8-character password has less entropy than a 56-bit key.

- Implication?
A question?

- Is there going to come a day where all passwords are useless?
  - Suppose I can remember 16 bytes of entropy (possible?)
  - Won’t there come a day when all passwords are useless?
    - Moore’s law and its corollaries?
Answer: no

• Nope, you just need to make the process of checking passwords more expensive. For example, you can repeat the salted hash many times ...

• Linear cost speedup?
Security in the News

• Security stories in the last 72 hours.
  • Equifax breach impacts 140+ million
  • Researchers Catch Microsoft Zero-Day Used To Install Government Spyware
  • BlueBorne Vulnerabilities Impact Over 5 Billion Bluetooth-Enabled Devices
  • ... and many more
• There is plenty to motivate your project selections for the semester!!
Something you have …

- Tokens (transponders, …)
  - Speedpass, EZ-pass
  - SecureID
  - Smartphones?
- Smartcards
  - Unpowered processors
  - Small NV storage
  - Tamper resistant
- Digital Certificates (used by Websites to authenticate themselves to customers)
  - More on this later …
A (simplified) sample token device

- A one-time password system that essentially uses a *hash chain* as authenticators.
  - For seed \((S)\) and chain length \((l)\), period \((x)\)
  - "Tamper proof" token encodes \(S\) in firmware
    
    \[
    pw_i = h^{l-i}(S)
    \]
  - Device display shows password for period \(i\)
  - Time synchronization allows authentication server to know what \(i\) is expected, and authenticate the user.

- *Note*: somebody can see your token display at some time but learn nothing useful for later periods.
Something you 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

• Issues with biometrics?
  • Revocation – lost fingerprint?
  • “Fuzzy” credential, e.g., your face changes based on mood ...
  • Great for physical security, challenges for on-line systems
**Biometrics Example**

- A fingerprint biometric device (of several)
- record the conductivity of the surface of your finger to build a “map” of the ridges
- scanned map converted into a graph by looking for landmarks, e.g., ridges, cores, ...
Fingerprint Biometrics (cont.)

- Graph is compared to database of authentic identities
- Graph is same, the person deemed “authentic”
  - This is a variant of the *graph isomorphism* problem
  - Problem: what does it mean to be the “same enough”
    - rotation
    - imperfect contact
    - finger damage

- *Fundamental*: False accept vs. false reject rates?
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 …

GET /protected/index.html HTTP/1.0

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

GET /protected/index.html HTTP/1.0
Authorization: Basic Sm9objpwYXNzd29yZA==
Setting up Basic auth in Apache

- File in directory to protect (.htaccess)

```
AuthType Basic
AuthName "None of your business"
AuthUserFile /cise/class/cnt5410fa17/public_html/secret/.htpw1
AuthGroupFile /dev/null
require valid-user
```

- In /usr/traynor/www-etc/.htpw1

```
  johndoe:$apr1$DOYdUsx7$tFJMFj80im2Bqdz/GFy8n1
```
generated using htpasswd program

- Can use different .htaccess files for different directories
- Can specify global policy in httpd.conf (if you have root on the server)
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. MD5(IP address:timestamp:server secret)

• Response: challenge hashed with user’s name & password
  ▶ 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
In-Class Exercise

• All students should have their laptops, with “John the Ripper” installed.

• Go to the course webpage for today’s lecture and download the password file “example.pwd”.

• Run “John” over this file and recover as many passwords as possible over the next 15 minutes.
Password Cracking Strategies

• What policies are enforced by organizations?

• Which is best?

What Does This Tell Us?

• In 20 minutes, what kinds of passwords were cracked?
  • Which changes made passwords stronger?
• Did the variants of “password” help?
  • Which changes were good? Bad?
• How do your adversary’s resources compare to yours?
• Exercise for home:
  • Create your own password file, containing the hashes of your UFL and Facebook accounts.
  • Run “John” for 24 hours.
Kerberos

• History: from UNIX to Networks (late 80s)
• Solves: password eavesdropping
• Online authentication
  • Variant of Needham-Schroeder protocol
• Easy application integration API
• First *single sign-on system* (SSO)
• Genesis: rsh, rcp
  • authentication via assertion

• Most widely used (non-web) centralized password system in existence (and lately only ...)
• Now: part of Windows Vista/7/8 network authentication
  • Old Windows authentication was extremely weak.
An aside …

- **Authentication**
  - Assessing identity of users
  - By using credentials …

- **Authorization**
  - Determining if users have the right to perform requested action (e.g., write a file, query a database, etc.)

- Kerberos authenticates users, but does not perform any authorization functions …
  - … beyond identify user as part of Realm
  - Typically done by application.

- Q: Do you use any “Kerberized” programs?
  - How do you know?
The setup . . .

• The players
  ▶ Principal - person being authenticated
  ▶ Service (verifier) - entity requiring authentication (e.g., AFS)
  ▶ Key Distribution Center (KDC)
    ▶ Trusted third party for key distribution
    ▶ Each principal and service has a Kerberos password known to KDC, which is munged to make a password ke, e.g., $k_A$
  ▶ Ticket granting server
    ▶ Server granting transient authentication

• The objectives
  ▶ Authenticate Alice (Principal) to Bob (Service)
  ▶ Negotiate a symmetric (secret) session key $k_{AB}$
The protocol

- A two-phase process

  1. User authentication/obtain session key (and ticket granting ticket) key from Key Distribution Center

  2. Authenticate Service/obtain session key for communication with service

- Setup

  - Every user and service get certified and assigns password
A Kerberos Ticket

- A kerberos ticket is a token that …
  - Alice is the only one that can open it
  - Contains a session key for Alice/Bob ($K^{AB}$)
  - Contains *inside it* a token that can only be opened by Bob

- Bob's Ticket contains
  - Alice's identity
  - The session key ($K^{AB}$)

- Q: What if issuing service is not trusted?

```
Ticket (K^{AB})

Ticket (K^{AB})
  “Locked” by $K^B$

“Locked” by $K^A$
```
The protocol (obtaining a TGT)

- $\text{Time}_{\text{exp}}$ - time of expiration
- $n$ - nonce (random, one-use value: e.g., timestamp)

$[A, TGS, \text{Time}_{\text{exp}}, n]$

$E(k^A, [k^A,TGS, TGS, \text{Time}_{\text{exp}}, n]), E(K^{TGS}, [A, k^A,TGS, \text{Time}_{\text{exp}}]),$
The protocol (performing authentication)

\[
[B, Time_{exp}, n, E(k^{A,TGS}, [B, Time_{exp}, n])], E(K^{TGS}, [A, k^{A,TGS}, Time_{exp}])]
\]

1. Alice

2. TGS

3. Bob

\[
E(k^{A,TGS}, [k^{A,B}, B, Time_{exp}, n]),
E(k^{B}, [A, k^{A,B}, Time_{exp}])
\]

\[
E(k^{A,B}, [A, Time_{exp}, n]),
E(k^{B}, [A, k^{A,B}, Time_{exp}])
\]

Authenticator
Cross-Realm Kerberos

- Extend philosophy to more servers
  - Obtain ticket from TGS for foreign *Realm*
  - Supply to TGS of foreign Realm
  - Rinse and repeat as necessary

- “There is no problem so hard in computer science that it cannot be solved by another layer of indirection.”
  - *David Wheeler, Cambridge University (circa 1950)*
Kerberos Reality

- V4 was supposed to be replaced by V5
- But wasn’t because interface was ugly, complicated, and encoding was infuriating
- Assumes *trusted path* between user and Kerberos
- Widely used in UNIX domains
- Robust and stable implementation
- *Problem*: trust ain’t transitive, so not so good for large collections of autonomous enterprises
Federated Identity

• Dealing with identity and authentication are challenging tasks.
  • How critical is identity management to the core of what you do?
  • How well does your company vet identity?
  • How well does it have to do so?
• Wouldn’t it be nice if I could simply use the same identity across a wide range of websites?
Single Sign On (SSO)

• With a single account/identity, you can log into multiple websites.

• Pros: Reduced password fatigue, the chance to widely deploy good policies …

• Cons: Single (or small # of) point(s) of failure, unintentional incompatibility issues.
OAuth

• OAuth is an *authorization* protocol

• Not an *authentication* protocol by itself.

• Participants include:
  
  • Relaying Parties (RPs): Web sites and apps
  
  • Identity Providers (IdPs): Facebook, OpenID

  • User Agents (UAs): Browsers
OAuth Message Flows

My App (RP) -> Bookface (IdP) -> User Agent

1 -> 2
3 -> 2
4
5

1
2
2
OAuth Problems

• OAuth is a framework, and not a protocol.
  • As such, there is lots of room to make mistakes.
    • Chen et al., OAuth Demystified for Mobile Application Developers, ACM CCS. 2014
  • Even reference implementations fail to follow the mandatory requirements specified in the RFC.
Wrap-Up

• Authentication establishes identity through the evaluation of credentials.

• Authentication Authorization
  • Getting this wrong often causes significant problems!

• Lots of complex protocols available to help achieve authentication.
  • Needham-Schroeder, Basic, Digest, Kerberos, OAuth*, and many more…

• Getting this correct is often the first step to securing your systems.