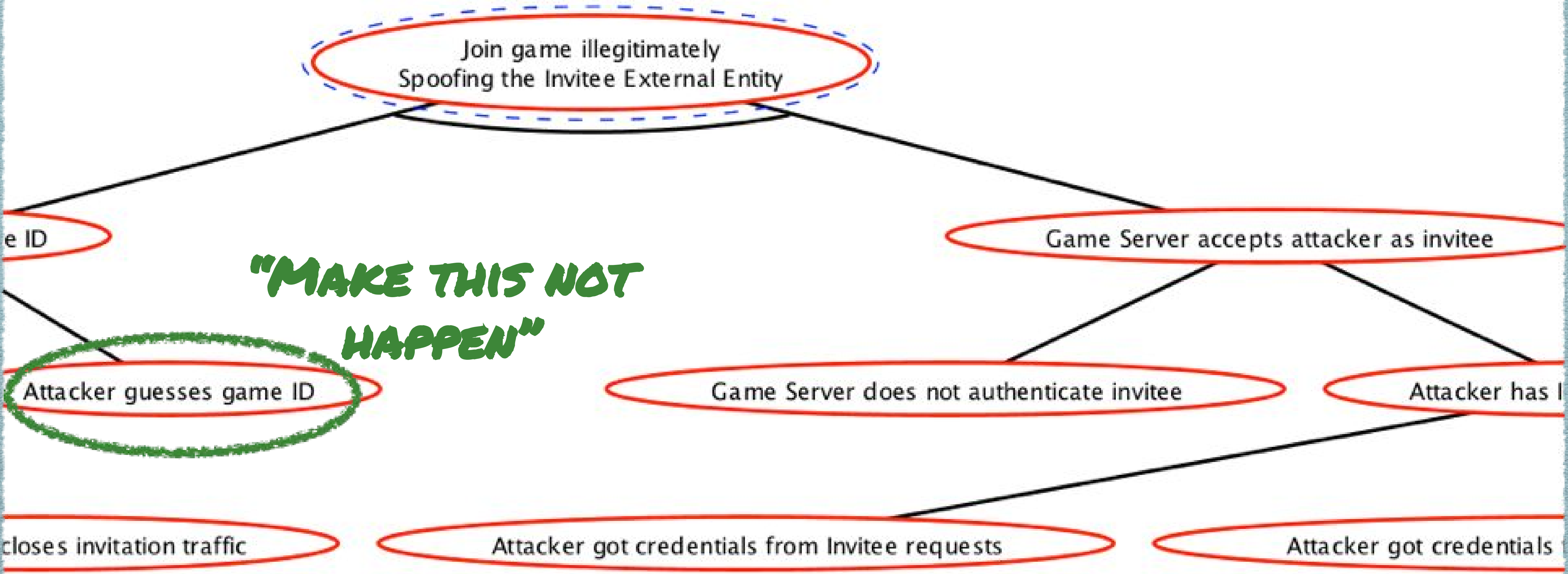
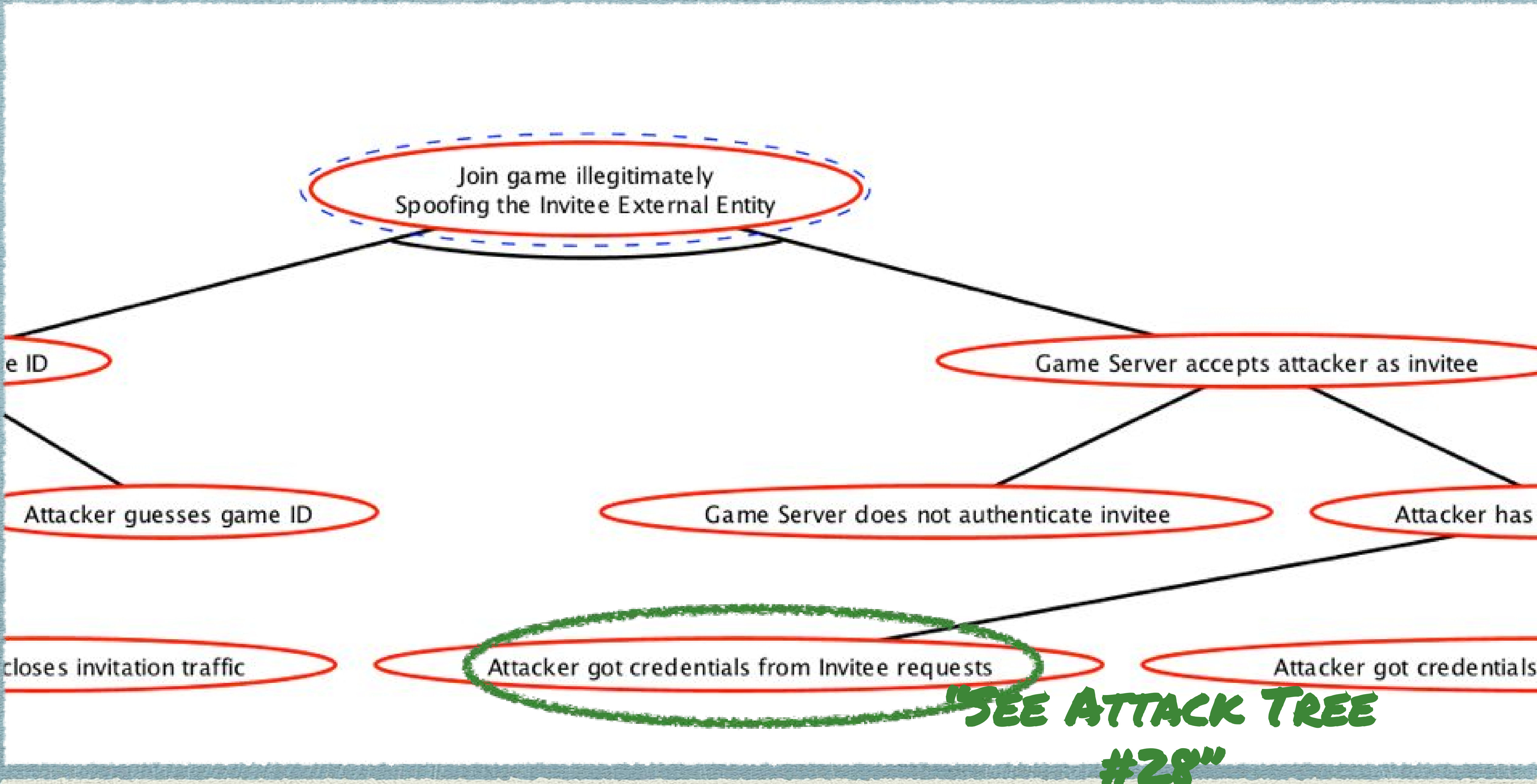
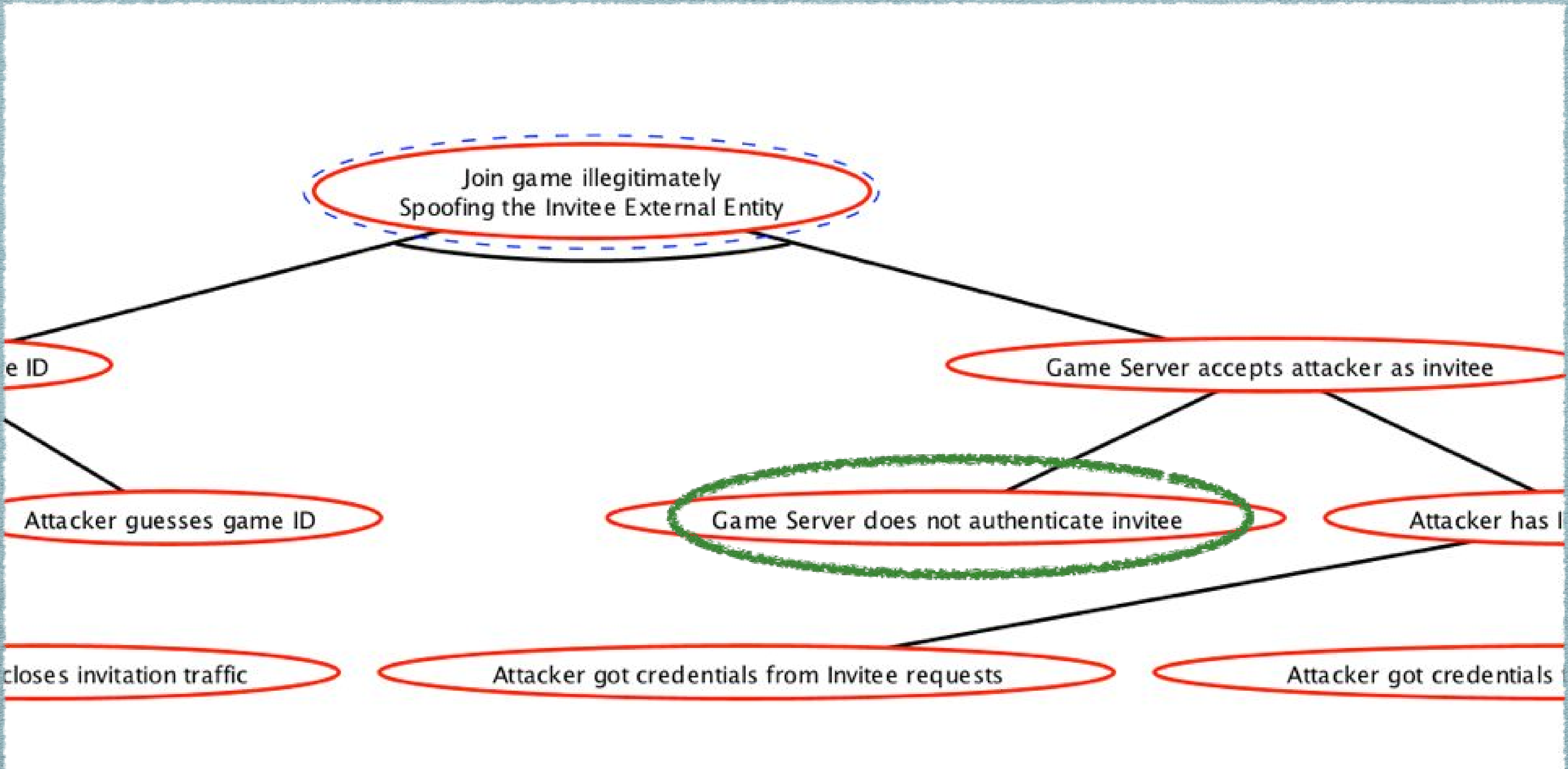
Mitigating Vulnerabilities



Sometimes it's self-evident



Sometimes it's another tree



Sometimes you're not sure

STRIDE → Fault

- Spoofing → lack of **identity**
- Tampering → lack of **integrity**
- Repudiation → lack of **auditing**
- Information disclosure → lack of **confidentiality**
- Denial of service → lack of **availability**
- Elevation of privilege → lack of **authorization**

Fault → Mitigation

- Identity → answer “who”
- Integrity → answer “who + what”
- Auditing → answer “who + what + when”
- Confidentiality → keep hidden
- Availability → eliminate waste
- Elevation of privilege → identity + integrity

Mitigation Toolbox

- Authentication
- Signing
- Logging
- Secrecy
- Filtering
- Defensive coding

Authentication

Determine whether an entity is in fact what it declares itself to be.

- Something it *knows*
- Something it *has*
- Something it *is*
- Also, chain of trust

REVOCABILIT
Y
&
DUPLICABILI
TY

Authentication

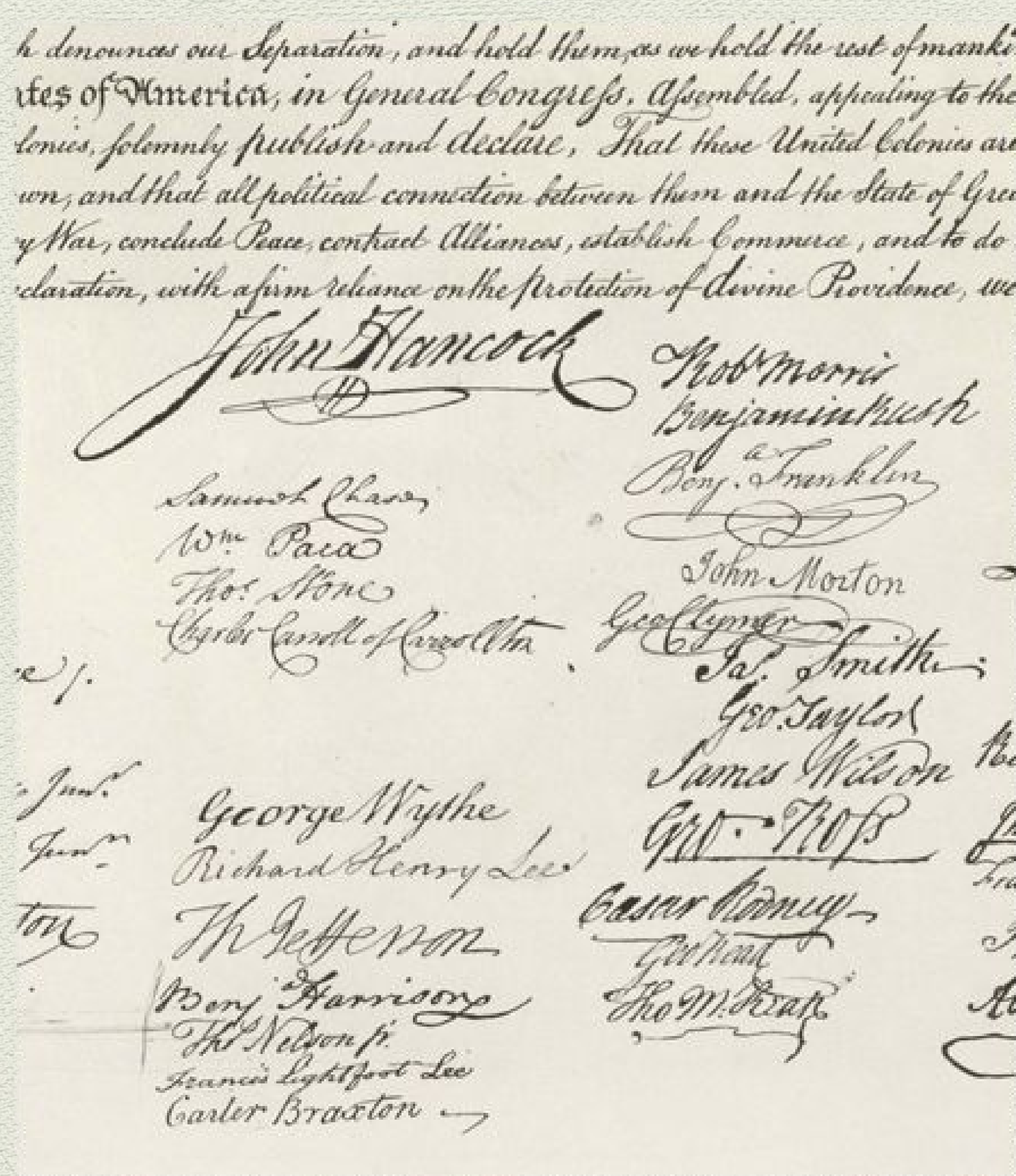
Demonstrate possession of a *shared secret* without revealing it

Signing

Determine whether a document is what an entity intended to send.

Hash

Shared secret



Hashes, Digests and Signatures

- A *hash* reduces an arbitrary document to a fixed-size representation
- A *digest* is a hash function where documents with even tiny differences are nearly guaranteed to have very different hashes
  - Ideally, also hard to invert
- A *signature* is a digest coupled with a shared secret

Cryptographic Hashes

- MD5, SHA-1, SHA-256, etc.
- Generally take a block (512 bits) of data, perform a series of shifts, rotates and XORs on the data, and repeat the process 20 to 40 times.
- Around 300,000,000 blocks a second on modern hardware.

Cryptographic Hashes

- The danger of relying on hashes is the chance of *collisions*
- The “birthday paradox” gives the basic chance of a collision as ~sqrt(keysize); weak hash algorithms offer ways to beat that
- Accidental collisions are almost impossible with any reasonable key size; we’re talking about the attacker finding a deliberate collision.

Replay Attacks

- Special kind of spoofing attack that involves replaying portions of a previous (valid) operation
- Mitigate with:
  - Time values
  - Message IDs or sequence numbers
  - Nonces

Nonces

- A value used on just one occasion
  - A temporary shared secret
- Generally a random number
  - But rand() won't cut it!
  - Even if you seed it from an unpredictable source
  - Cryptographically-secure RNGs generate unpredictable results… and take longer, natch

Logging

Log important activity for later review.

- Authenticated users
- Client keys
- IP addresses

Encryption

Encodes a document so that only authorized parties can read it.

- Symmetric: The same key encrypts and decrypts
- Asymmetric: There are two keys, and each decrypts what the other encrypts

Symmetric Ciphers

- DES, 3DES, AES (formerly known as Rijndael)
- Generally take a block (128 bits) of data, XOR it against a key, mutate the block and key and XOR again… for 10 or 15 rounds.
- Around 200,000,000 blocks a second on modern hardware.

Symmetric Key Size

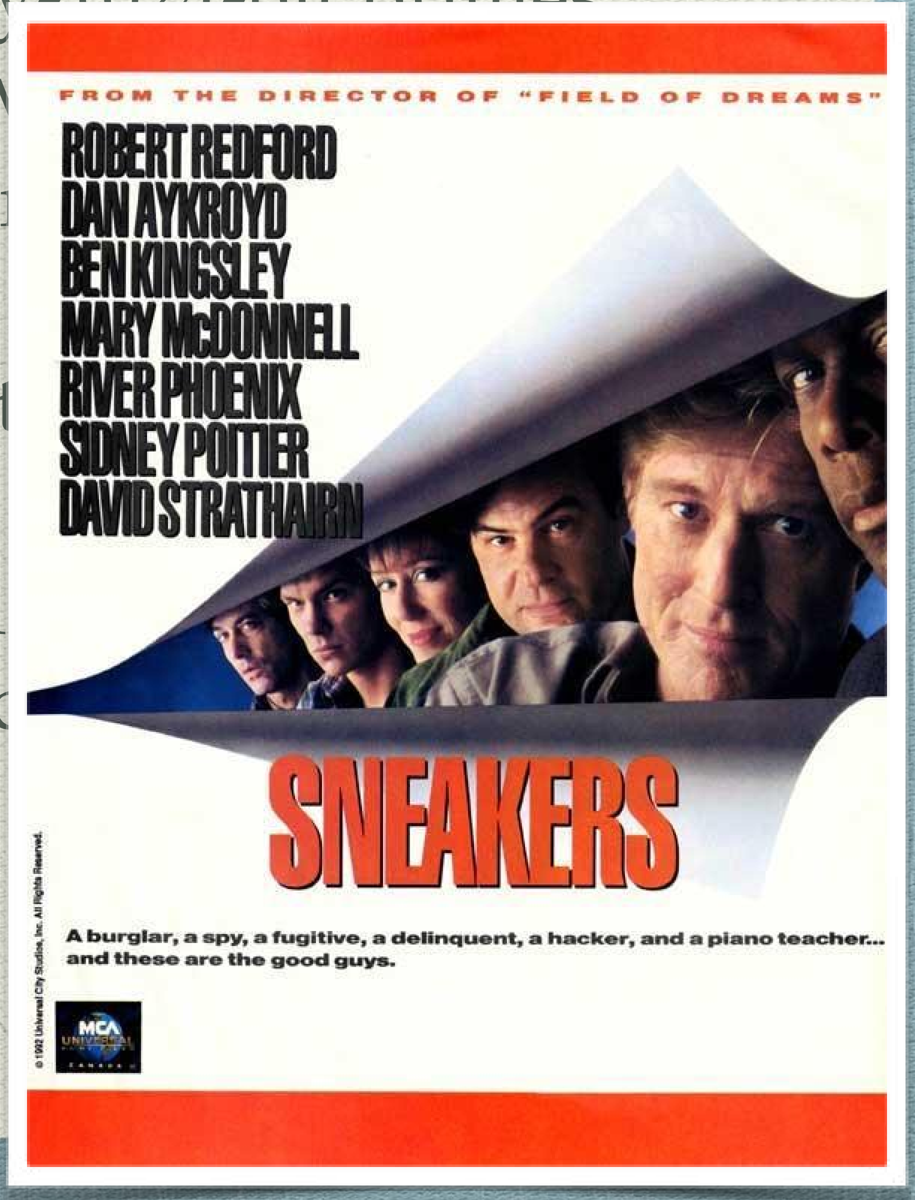
- A cipher is considered secure if there is no better attack than brute force.
- There are 18,500,000,000,000,000,000 64-bit keys.
  - At 200M a second, that'd take 2,933 years.
  - *Or* it'd take 1M computers one day.
- So we use 128-bit keys.
  - When quantum computing gets solved, we'll have to switch to 256-bit keys.

Asymmetric Ciphers

- RSA, Elliptic Curve Cryptography
- Encryption and decryption involve raising very large numbers to the power of other very large numbers… not cheap, and not suitable for large data.
- Around 1,500 blocks a second on modern hardware.

Asymmetric Key Size

- An RSA key is derived from two prime numbers multiplied together. A "2048-bit key" is a pair of roughly-1024-bit primes and if you know one you automatically know
- There are a lot of 2048-bit numbers… but ther many 1024-bit primes.
  - There's about as many 1,624-bit primes as t numbers.
- 1024-bit keys are a few years away from bein 2048-bit RSA keys are okay for now, but the c
  - Constant risk that someone will discover a door" to factor large numbers quickly…



Key Exchange

- Alice generates a random 128-bit AES key.
- Alice encrypts it with Bob's public RSA key.
- Alice sends the encrypted AES key to Bob.
- Bob decrypts the AES key with his private RSA key.
- Bob and Alice now have a shared secret.
    - …as long as it's really Bob and Alice.
    - …and as long as it's really secret.

Filtering

Reject invalid or abusive activity as efficiently as possible.

- Be careful this doesn't leak information…
- …or deny service to legitimate users

Defensive Coding

Ensure continuing safe function despite unexpected or malicious input.

Defensive Coding Techniques

- Handle errors
  - Beware information disclosure in error messages
- Fail “safe”
- Watch out for:
  - Buffer overruns
  - Integer overflows

Mitigation Toolbox

- *Authentication* via shared secrets
- *Signing* via cryptographically-secure hashes of data combined with shared secrets
  - Beware of replay attacks and weak RNGs!
- *Auditing* via logging
- *Encryption* via symmetric or asymmetric ciphers
- *Filtering* to impose reasonable limits on data flows
- *Defensive coding* practices such as input validation

Five by Five

1. It’s easy to inject packets into the middle of a UDP conversation. It’s harder to do with a TCP conversation. Explain why that is, using terms from this deck. What kind of STRIDE attack would this represent?

Five by Five

2. The checksum in UDP and TCP headers is a simple form of ___________, and thus can be considered a mitigation against some trivial __________ attacks. Describe briefly how an attacker could defeat it. How would a defender mitigate that, and what would it cost?

Five by Five

3. Replay attacks are a form of ________. List three things you could include in a packet or request to defend against replay attacks.

Five by Five

4. Filtering is a defense against _________ attacks. If poorly implemented, however, it can itself become a vector for _________ and _________ attacks. Give a one-sentence example of each of those two attacks against a filtering mechanism.

Five by Five

5. The old standard for encrypting browser connections was called _____; the new version is called _____. Both will reliably encrypt a TCP stream. However, simply encrypting the stream isn't sufficient—the client must also protect against _______ attacks by doing what?