Using A Cost-Based Framework For Analyzing Denial Of Service

Presented By: Joan Paul

- A Cost-Based Framework for Analysis of Denial of Service in Networks – Catherine Meadows (2001)
- Analyzing DoS-Resistance of Protocols Using a Cost-Based Framework – Vijay Ramachandran (2002)
- Modelling Denial of Service Attacks on JFK with Meadows's Cost-Based Framework – J. Smith, J.M. Gonzales-Nieto, C. Boyd (2006)

Denial of Sevice (DoS)

- Aims to exhaust the processing, memory, or network resources of target systems
- Solutions/mitigations
  - increase defender's resources
  - reduce defender's cost of servicing a request
    - reduce memory storage cost – state maintained by initiator
    - reduce processing cost – have initiators aid the responder in doing expensive operations
  - increase cost of making a request – puzzles
  - assuring origin of requests – cookies

The Framework

- views DoS as a resource exhaustion problem
- cost-based, so capable of expressing DoS resistance in a quantifiable manner
- mostly applicable to cryptographic protocols, which uses most expensive form of authentication
- employs formal methods

Analyzing a Protocol's DoS-Susceptibility

- Show that certain properties hold at each step of the protocol
- Intruder's strengths may vary as protocol progresses

As compared to analyzing a protocol for authentication properties:

- Prove its requirements are satisfied when protocol completes
- Prove protocol is sound against a uniformly strong intruder

In the end, we would like to know whether or not the protocol allows the server/responder(potential victim) to be available to participate in a protocol execution with legitimate clients/initiators, even in the face of active attackers.

Fail-stop Protocols

- The cost-based framework is based on the notion of a fail-stop protocol
  - fail-stop - halts upon the detection of any message that has been interfered with (replay, manufactured by intruder, out-of-sequence)

- Share desirable properties with DoS-resistant protocols

- Tend to use strong authentication up-front, making it vulnerable to DoS attacks

- Concept needs modification to make it applicable, by incorporating actions performed in a protocol execution and the cost associated with them.

Protocol Specification

Annotated Alice-and-Bob specification *P* is a sequence of statements of the form:

$$L : A \rightarrow B: T_1, \ldots, T_k \;||\; M \;||O_1, \ldots, O_n$$

Example:

$L1.\ I \rightarrow R : computenonce_1(N_I),\ N'_I{=}hash_1(N_I),\ createexp_1(g^i)\ ||$
$\qquad N'_I\ ,\ g^i\ ||$
$\qquad verifygroup(g^i),\ accept_1$

Cost Sets and Cost Functions

- Cost set C is a monoid with operator + and partial order $\leq$ s.t. $x \leq x + y$ and $y \leq x + y$, $\forall x, y \in C$.

  C : { 0 < cheap < medium < expensive }

  cheap + medium = medium

- Event-cost function δ maps events to a cost set C and is 0 on accept events.

  δ(computenonce) = cheap, δ(accept) = 0

Cost Sets and Cost Functions

- A message-processing cost function, δ', is defined on verifications events $\{V_i\} \subset \{O_j\}$ s.t. for $A \rightarrow B$: ...|| M || $O_1, \ldots, O_n$, if $V_i = O_j$, then $\delta'(V_i) = \delta(O_1) + \ldots + \delta(O_j)$.

$$\delta'(\text{verify}_2) = \delta(\text{verify}_1) + \delta(\text{verify}_2)$$

- A protocol-engagement cost function, Δ, is defined on accept event O_n s.t. $\Delta(O_n)$ is the sum of all costs of operations at the receiver up to O_n, plus the costs of any immediate message preparations

$$\Delta(\text{accept}_1) = \delta(\text{verify}_1) + \delta(\text{verify}_2) + \delta(\text{compute}_3)$$

L1. $I \rightarrow R$: $\text{compute}_1(X_1)$, $\text{compute}_2(X_2)$ || X_1, X_2 ||
$\text{verify}_1(X_1)$, $\text{verify}_2(X_2)$, accept_1

L2: $I \leftarrow R$: $\text{compute}_3(Y_1)$ || Y_1 || $\text{verify}_3(Y_1)$, accept_2

Intruder Cost Functions

- Let G be the attacker cost set, and I be the set of intruder actions. The function ϕ maps intruder actions to their costs in G.

- The intruder cost function Φ is defined on a sequence of attacker actions as $\Phi(\{i_1, \ldots, i_n\}) = \phi(i_1) + \ldots + \phi(i_n)$ for $i_k \in I$.

Modified Fail-stop

- The attack cost function, Θ, maps events from specification P to a cost set C.
  P is fail-stop with respect to Θ, if for every event $E \in P$, no events occur after E, unless the cost to the attacker is at least $\Theta(E)$.

- Let C and G be the responder and the attacker cost sets respectively.
  A tolerance relation T is the subset of C x G that consists of all pairs (c, g) s.t. the defender will expend cost c only if the attacker will expend resources of at least cost g.
  A tuple (c', g') is said to be within the tolerance relation if there exists $(c, g) \in T$, s.t. $c' \leq c$ and $g' \geq g$.

Tolerance Relations

- (0, 0), (cheap, cheap), (medium, medium), (expensive, expensive) - acceptable
- (cheap, medium), (medium, expensive) – more restrictive
- (medium, cheap) – more tolerant
- (expensive, cheap) – unacceptable

General Steps for Evaluating a Protocol's Susceptibility to DoS

1. Decide what your cost function is and what you assume to be the intruder's capabilities

2. Decide what your tolerance relation is

3. Determine the attack cost function, Θ for each step of the protocol

4. For each attack cost function in 3, determine that:
   a. if event E_1 is immediately preceding a verification event E_2, then $(\delta'(E_1), \Theta(E_2)) \in T$
   b. if E is an accept event, then $(\Delta(E), \Theta(E)) \in T$

Just Fast Keying (JFK) Protocol

1. $I \rightarrow R : N_I', g^i, ID_R'$
2. $I \leftarrow R : N_I', N_R, g^r, grpinfo_R, ID_R, S_R[g^r, grpinfo_R], token$
3. $I \rightarrow R : N_I, N_R, g^i, g^r, token, \{ID_I, sa, S_I[N_I', N_R, g^i, g^r, ID_R, sa]\}_{K_a}^{K_e}$
4. $I \leftarrow R : \{S_R[N_I', N_R, g^i, g^r, ID_I, sa, sa'], sa'\}_{K_a}^{K_e}$

$$N_I' = H(N_I)$$
$$token = H_{HK_R}(g^r, N_R, N_I', IP_I)$$
$$K_e = H_{g^{ir}}(N_I', N_R, \text{“1”})$$
$$K_a = H_{g^{ir}}(N_I', N_R, \text{“2”})$$
$$K_{ir} = H_{g^{ir}}(N_I', N_R, \text{“0”})$$

Table 1: Just Fast Keying (JFK) Message Components

$S_x[M]$	digital signature of M with private key of x.
$H(M)$	an unkeyed hash of M.
$H_K(M)$	keyed hash of M with key K.
$\{M\}_{K_a}^{K_e}$	encryption of M under key K_e followed by the generation of a message authentication code (MAC) over the resultant ciphertext with key K_a.
IP_I	initiator's network address.
g	Generator of a multiplicative group of order q.
r, i	Integers between 1 and q chosen at random by R and I.
g^i, g^r	initiator and responder current exponential
N_I, N_R	Random nonces chosen by I and R.
N_I'	initiator's initial nonce, computed as $H(N_I)$.
ID_I, ID_R	Information identifying I and R's public keys (certificates).
ID_R'	Hint to R of identity I wants to establish communications with.
HK_R	Transient, secret hash key used by responder.
sa, sa'	Information used in establishing a security association.
$grpinfo_R$	Groups supported by responder, algorithms and hash functions.

Annotated Alice-and-Bob Specification of JFK

L1. $I \rightarrow R$: computenonce$_1$(N_I), N'_I=hash$_1$(N_I), createexp$_1$(g^i) ||
N'_I , g^i || verifygroup(g^i), accept$_1$

L2: $I \leftarrow R$: computenonce$_2$(N_R), token=generatemac$_1$(K_R, {g^r, N_R, N'_I, IP_I}), ||
N'_I , N_R, g^r, groupinfo$_R$, ID_R, S_R{g^r, groupinfo$_R$}, token ||
verifysig$_1$, accept$_2$

L3: $I \rightarrow R$: generatedh$_1$(g^{ir}), K=computekeys$_1$(N'_I , N_R, g^{ir}),
T=generatesig$_1$(N'_I , N_R, g^i , g^r, ID_R, sa), C'=encrypt$_1$(K, {ID_I, T, sa}),
C=generatemac$_2$(K, C') || N'_I , N_R, g^i , g^r, token, C, C' ||
N'_I=hash$_2$(N_I), verify$_1$(token=generatemac$_3$(K_R, {g^r, N_R, N'_I, IP_I}),
generatedh$_2$(g^{ir}), K=computekeys$_2$(N'_I, N_R, g^{ir}),
verify$_2$(C=generatemac$_4$(K, C')), decrypt$_1$(K, C'), verifysig$_2$(T), accept$_3$

L4: $I \leftarrow R$: W=generatesig$_2$(N'_I , N_R, g^i , g^r, ID_I, sa, sa'), D'=encrypt$_2$(K,{W, sa'}),
D=generatemac$_5$(K, D') || D', D ||
verify(D=generatemac$_6$(K, D')), decrypt$_2$(K, D'), verifysig$_3$(W), accept$_4$

Applying the Framework on JFK

- C and G: { 0 < cheap < medium < expensive }
- T = { (cheap, cheap), (cheap, medium), (cheap, expensive), (medium, cheap), (medium, medium), (medium, expensive), (expensive, expensive) }
- Events and associated costs:
  - δ(computenonce) = cheap
  - δ(hash) = cheap
  - δ(createexp) = expensive
  - δ(verifygroup) = medium
  - δ(generatemac) = medium
  - δ(generateh) = expensive
  - δ(computekeys) = medium
  - δ(generatesig) = expensive
  - δ(verifysig) = expensive
  - δ(en/decrypt) = medium

JFK Analysis – Evaluation of Costs

Evaluation up to event accept_1 :

L1. $I \rightarrow R$: $\text{computenonce}_1(N_I)$, $N'_I = \text{hash}_1(N_I)$, $\text{createexp}_1(g^i)$ ||
N'_I , g^i || $\text{verifygroup}(g^i)$, accept_1

- $\Theta(\text{accept}_1)$ = cheap, since createexp could be spoofed and $\phi(\text{spoofexp})$ = cheap

- $\Delta(\text{accept}_1) = \delta(\text{verifygroup}) + \delta(\text{computenonce}_2) + \delta(\text{generatemac}_1)$ = medium

$(\Delta(\text{accept}_1), \Theta(\text{accept}_1))$ = (medium, cheap) $\in T$

JFK Analysis – Evaluation of Costs

Evaluation up to $accept_2$:

L2: $I \leftarrow R$: $computenonce_2(N_R)$,
$token = generatemac_1(K_{R,} \{g^r, N_{R,} N'_I, IP_I\})$, ||
N'_I, N_R, g^r, $groupinfo_R$, ID_R, $S_R\{g^r, groupinfo_R\}$, token ||
$verifysig_1$, $accept_2$

- $\Delta(accept_2) = \delta(verifygroup) + \delta(computenonce_2) + \delta(generatemac_1)$
  = medium + cheap + medium
  = medium

- $\Theta(accept_2)$ = cheap, since spoofing exponent from L_1 is cheap and $\phi(accept_2) = 0$, and attacker need not do an actual $verifysig_1$ which is normally expensive

$(\Delta(accept_2), \Theta(accept_2)) = (medium, cheap) \in T$

JFK Analysis – Evaluation of Costs

Evaluation up to $accept_3$:

L3: $I \rightarrow R$: $generatedh_1(g^{ir})$, $K=computekeys_1(N'_I, N_R, g^{ir})$,
$T=generatesig_1(N'_I, N_R, g^i, g^r, ID_R, sa)$, $C'=encrypt_1(K, \{ID_I, T, sa\})$,
$C=generatemac_2(K, C') \,||\, N'_I, N_R, g^i, g^r, token, C, C' \,||$
$N'_I=hash_2(N_I)$, $verify_1(token=generatemac_3(K_R, \{g^r, N_R, N'_I, IP_I\}))$,
$generatedh_2(g^{ir})$, $K=computekeys_2(N'_I, N_R, g^{ir})$,
$verify_2(C=generatemac_4(K, C'))$, $decrypt_1(K, C')$, $verifysig_2(T)$, $accept_3$

Message processing costs:

- $\delta'(verify_1)$ = medium, resulting in a tolerance relation (medium, cheap) and $(\delta'(verify_1), \Theta(\text{receive msg } 3)) \in T$
- $\delta'(verify_2)$ = expensive, since responder must do exponentiation and key derivation before message authentication can be verified

JFK Analysis – Evaluation of Costs

Message processing cost (contd.):

- Θ(verify$_1$) = cheap, since spoofing C and C' is cheap, so:
  (δ'(verify$_2$), Θ(verify$_1$)) = (expensive, cheap) $\notin T$
  which means possible DoS attack on the protocol

- δ'(verifysig$_2$) = expensive
  Θ(verify$_2$) = expensive, since attacker must construct message that passes verify$_1$ and verify$_2$ so:
  (δ'(verifysig$_2$), Θ(verify$_2$)) $\in T$

Protocol engagement costs:

- Δ(accept$_3$) = expensive, this includes message generated in L_4

- Θ(accept$_3$) = expensive, so: (Δ(accept$_3$), Θ(accept$_3$)) $\in T$

Framework Limitations

How about distributed denial of service(DDoS)?
Modify the application of the framework by:

- determine precise relationships between elements in cost set
  - medium cost = two cheap events
  - expensive event = three medium cost events

- identifying the computational events whose results can be reused and represent costs of those events with a fractional modifier *n*, number of nodes over which the event is distributed
  - ϕ(createexp) = expensive
  - ϕ(shareexp) = (1/n) * expensive = cheap (for larger n)
  - ϕ(shareexp) = (1/n) * expensive = medium (for smaller n)

Other Limitations of the Framework

- the need for more refined, realistic, and sensitive cost functions
  - comparing difficulty of two distinct operations
  - may not be interested in just cost but its ratio to available resources

- attacker's capabilities do not always equal defender's capabilities
  - assumptions have to be made about what the attackers are capable of

- does not address bandwidth exhaustion

- application of the framework for protocol analysis is not automated

Applicability of the Framework to Existing Tools and Models

- Could possibly modify and use tools that use state exploration techniques (FDR/Casper and Murφ, Interrogator, NRL) since standard intruder model is part of the tool

- Could possibly use high-level protocol description languages (CAPSL, Casper) since
  - these are based on Alice-and-Bob notation
  - translators for most of these languages infer the operations directly from specification, so just need to add an estimate cost of each type of operations

Q & A

Functions and Definitions

An Alice-and-Bob specification is a sequence of statements of the form $A \rightarrow B$: M where M is the message sent from A to B.

Annotated Alice-and-Bob specification *P* is a sequence of statements of the form:

$$L : A \rightarrow B: T_1, \ldots, T_k \,||\, M \,||\, O_1, \ldots, O_n$$

$T_1, \ldots, T_k$ – ordered steps taken by A to produce M

$O_1, \ldots, O_n$ – ordered steps taken by B to process and verify M

Functions and Definitions

Let L_i : $A \rightarrow B$: $T_1, \ldots, T_k$ || M ||$O_1, \ldots, O_n$ be the ith line in an annotated Alice-and-Bob specification. X is an event in L_i if:

1. X is one of T_i or O_j
2. X is a "A sends M to B" or "B receives M from A"

Events T_i and "A sends M to B" are said to occur at A, and events O_j and "B receives M from A" are said to occur at B.

Types of events:

- normal – always succeed, occur at sender or receiver
- verification – may succeed or fail, occur only at receiver
- accept – reserved event, O_n, that only occurs at the receiver

Modelling DDoS in Cost-Based Framework

- Consider *n* coordinated attackers, generating a single g^i resulting in an event cost for createexp to be amortized over all attackers (i.e. ϕ(shareexp) = (1/n) * expensive)

- g^{ir} in message three can also be computed once and distributed (i.e. ϕ(sharedh) = (1/n) * expensive)

- For smaller values of n,
  ϕ(shareexp) = ϕ(sharedh) = medium, and
  ϕ(shareexp) = ϕ(sharedh) = cheap, for larger values of n

Possible JFK DDoS Attack

- Attackers will want responder to perform the expensive signature verification in message three, requiring generation of valid messages up to and including $decrypt_1$.

- In constructing message three, attackers have event cost function equivalent to legitimate protocol participants except:
  - ϕ(sharedh) = (1/n) * expensive (medium for smaller n)
  - ϕ(spoofsig) = cheap

Hence:

$$\Theta(decrypt_1) = \phi(sharedh) + \delta(computekeys_1) + \phi(spoofsig) + \delta(encrypt_1) + \delta(generatemac_2) = medium$$

Message Processing Cost Calculation

- Message processing cost (δ') to responder in order to verify that message three is bogus include:
  $\delta'(verifysig_2) = \delta(hash_2) + 2 * \delta(generatemac) +$
  $\delta(generatedh_2) + \delta(computekeys_2) +$
  $\delta(decrypt_1) + \delta(verifysig_2)$

- Dominated by expensive costs resulting in a tolerance relation:
  $(\delta'(verifysig_2), \Theta(decrypt_1)) = (expensive, medium) \notin T$, a possible DoS attack