プロトコルの脆弱性

(株)インターネットイニシアティブ
永尾 禎啓　nagao@iij.ad.jp

TCP/IP プロトコルスタックの脆弱性

- プロトコルの仕様から見た脆弱性の分類
  1. 仕様は正しいが、実装上のバグ
  2. 仕様の曖昧さに起因
     - 実装によっては脆弱性が存在
  3. 仕様自体のバグ
  4. バグではないが仕様上不可避な問題
- 「プロトコルの脆弱性」とは
  - プロトコルの仕様に起因する脆弱性
  - 1. は含まれない

TCP/IP プロトコルの脆弱性

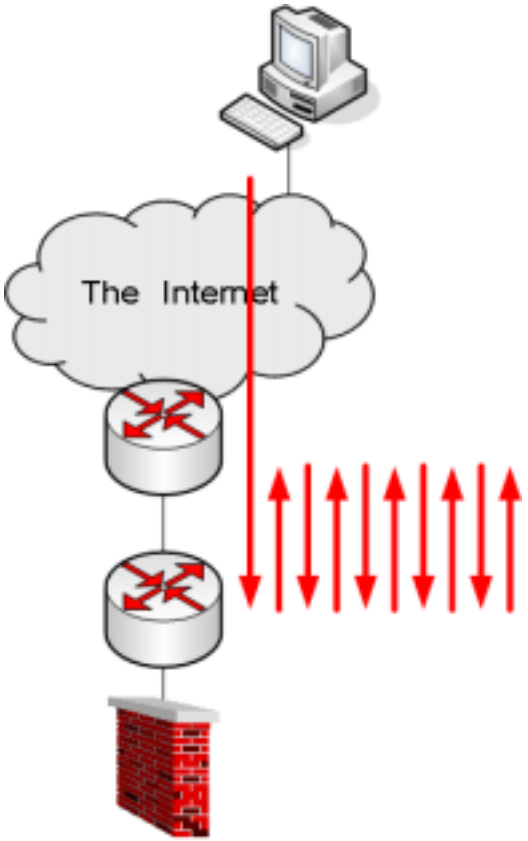
- IP の脆弱性
- ICMP の脆弱性
- UDP の脆弱性
- TCP の脆弱性

IP の脆弱性

- 仕様: RFC791
- IP オプション
  - ソースルートオプション
- フラグメンテーション/リアセンブル
  - Ping of Death 攻撃
  - Teardrop 攻撃
  - 大量フラグメント攻撃

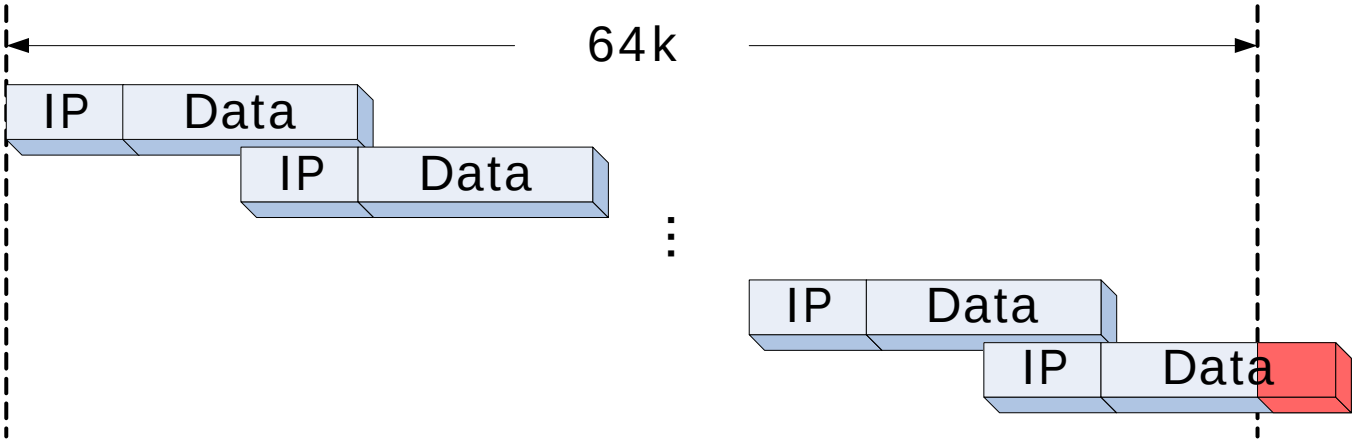
IP の脆弱性 ～ ソースルートオプション

- LSRR オプションで往復させる
  - 送出トラフィックの5倍 x 2
  - Ping パケットならさらに戻りも
- 仕様を厳格に守る限り不可避
- 対策
  - 不要な IP オプションをブロック



IP の脆弱性 ～ Ping of Death 攻撃

- リアセンブル後の長さが 65535 を超える
  - フラグメント offset + length の異常
  - 実装によってはクラッシュ/リブート/ハングする
    - OS 内部の計算でマイナスになる
    - コピー先として確保したメモリをはみ出す
  - ping –l 65510 <host> (Windows95)

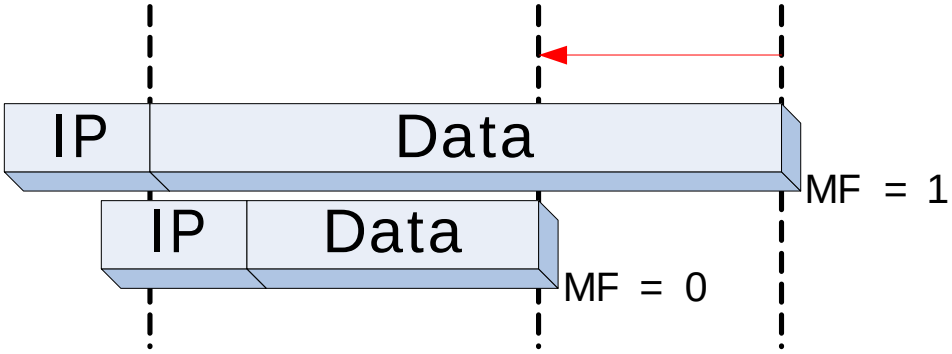


IP の脆弱性 ～ Ping of Death 攻撃

- 仕様に記述がない異常時の取り扱いに起因
- 現在の OS は修正済み

IP の脆弱性 ～ Teardrop 攻撃

- 最終フラグメントが前のフラグメントより短い
  - 重複フラグメントの length 計算の異常
  - 実装によってはクラッシュ/リブート/ハングする
    - OS 内部の計算でマイナスになる
    - コピー先として確保したメモリをはみ出す

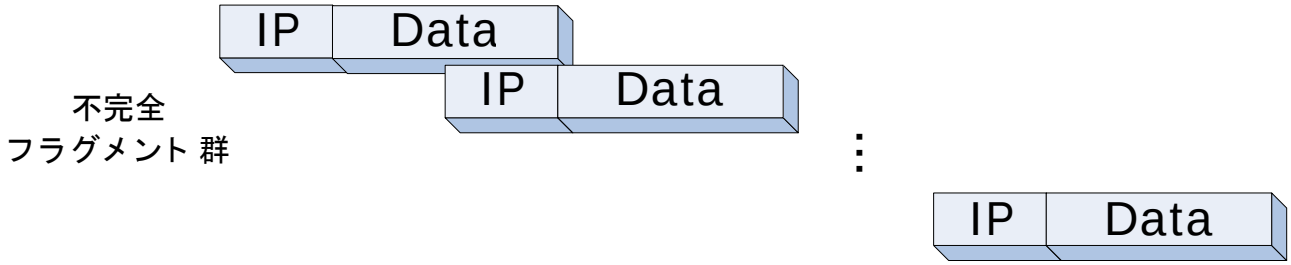


IP の脆弱性 ～ Teardrop 攻撃

- 仕様に記述がない異常時の取り扱いに起因
- 現在の OS は修正済み

IP の脆弱性 ～ 大量フラグメント攻撃

- 大量のフラグメントを送りつける
  - リアセンブル可能になるまでフラグメントを保持
  - 仕様ではフラグメントを最短でも TTL 秒保持する
  - mbuf が枯渇する可能性
    - ほとんど/まったく不応答になる
    - DoS



IP の脆弱性 ～ 大量フラグメント攻撃

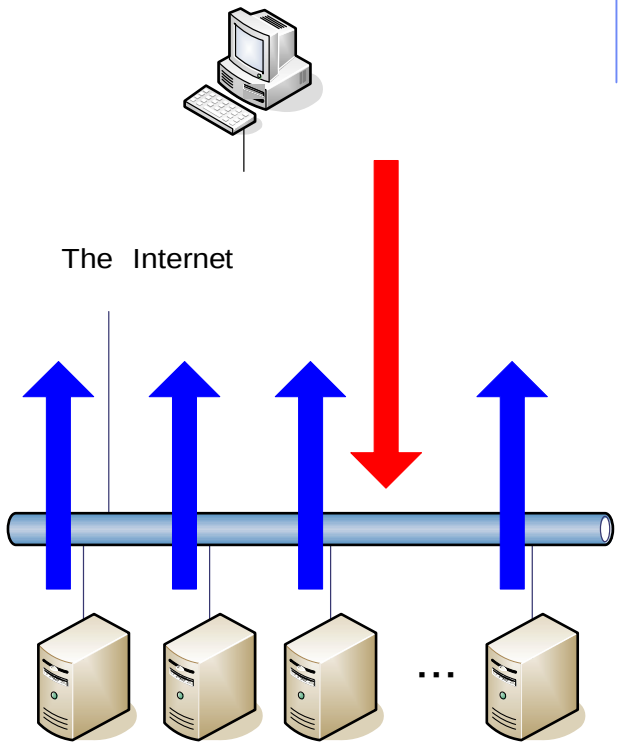
- 仕様を厳格に守る限り不可避
- 対策
  - 現在の OS では、保持するフラグメントの最大個数を設けている
  - TTL 秒経過しなくても破棄する

ICMP の脆弱性

- 仕様: RFC792
- 診断メッセージ
  - Smurf 攻撃
- エラー通知メッセージ
  - 偽のエラー通知による DoS
    - Unreachable, Needs Fragment など

ICMP の脆弱性 ～ Smurf 攻撃

- ブロードキャスト宛 ping (Echo Request)
  - 他の診断系メッセージも同様
- Smurf Amplifier Registry
  http://www.powertech.no/smurf/

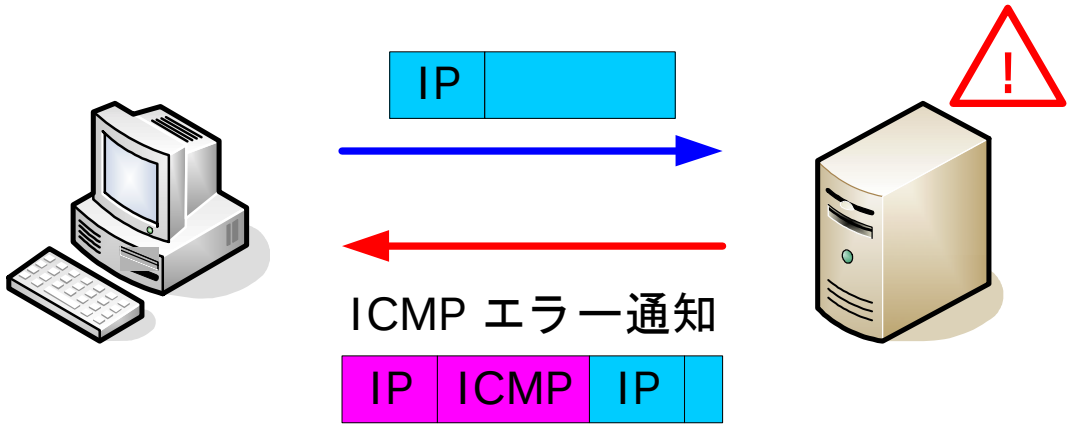


ICMP の脆弱性 ～ Smurf 攻撃

- 仕様に記述がない宛先の取り扱いに関する脆弱性
- 対策
  - broadcast 宛 ICMP をブロック
  - 現在の対外の OS では、デフォルトで broadcast 宛 ping に答えない

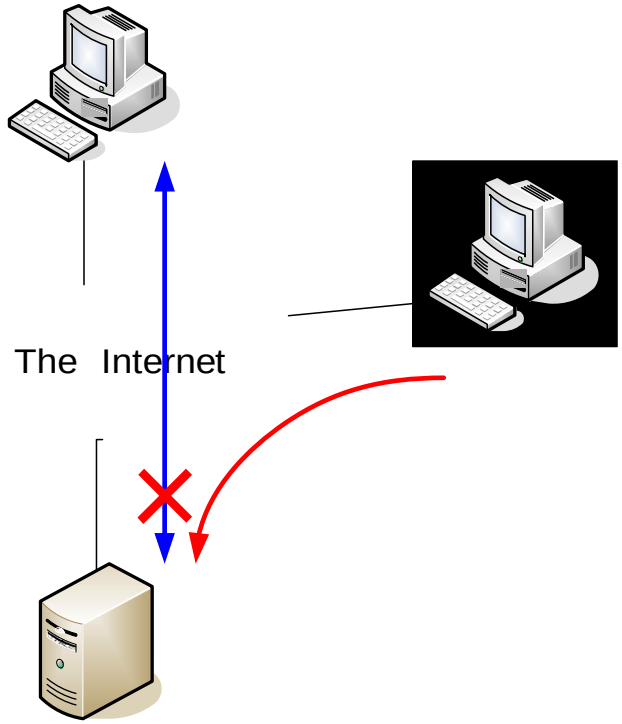
ICMP の脆弱性 ～ 偽のエラー通知による DoS

- ICMP エラー通知
- 受け取ったパケットに問題があるとき返送
  - Unreachable (Port, Host, Net, Protocol, ...)
  - Needs Fragment
  - など



ICMP の脆弱性 ～ 偽のエラー通知による DoS

- 偽造されたエラー通知
  - Unreachable による切断
  - Needs Fragment による PMTU 低下
- ただし、IP アドレスとポートの情報が必要



ICMP の脆弱性 ～ 偽のエラー通知による DoS

- 仕様上不可避
- 対策
  - アドレス・ポート推測のために大量 ICMP エラー通知が送られてくるのを検知するくらいか

UDP の脆弱性

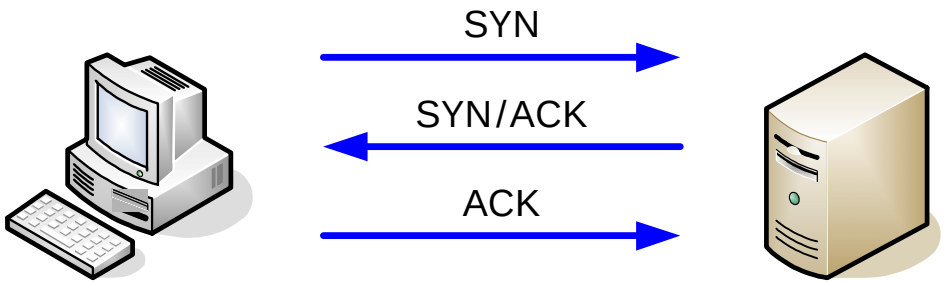
- 仕様: RFC768
- UDP = IP + ポート + チェックサム
- 「UDP による攻撃」と言われるものでも、たいてい本質は IP の脆弱性によるもの
  - 例: Teardrop 攻撃
  - 流布した攻撃ツールがたまたま UDP を使っていただけ

TCP の脆弱性

- 仕様: RFC793
- 3-way ハンドシェイク
  - SYN flood 攻撃
  - ISN 推測攻撃
- セグメンテーション/リアセンブル
  - 大量不完全セグメント攻撃

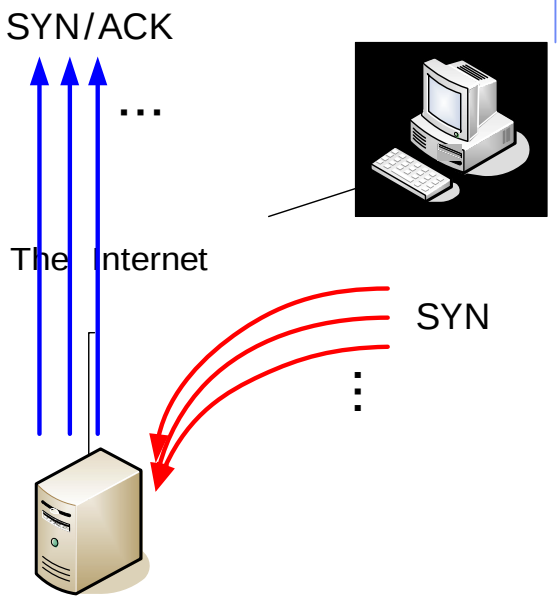
TCP の脆弱性 ～ SYN flood 攻撃

- 通常は SYN に SYN/ACK が返り、さらに ACK を送ることで 3-way ハンドシェイクが完了する



TCP の脆弱性 ～ SYN flood 攻撃

- SYN を大量に送りつける
- SYN/ACK に ACK 返答もする型
  - 攻撃側もリスクが大きい
    - 攻撃側にリソースが必要
    - アドレスの spoof ができない
- SYN/ACK に ACK 返答しない型

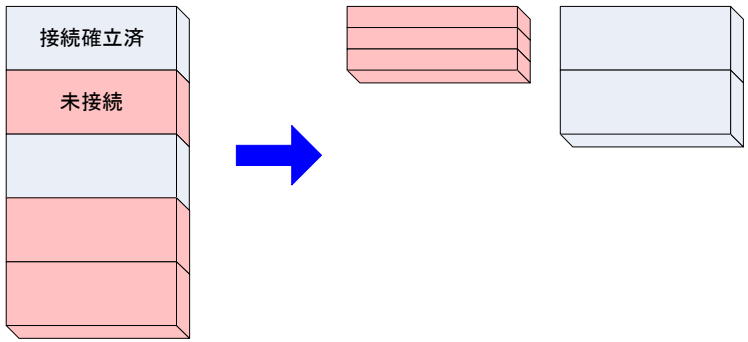


TCP の脆弱性 ～ SYN flood 攻撃

- 影響
  - サーバ OS のメモリ消費
  - 上限に達すると、それ以上接続を受けられない
  - DoS
- 仕様上不可避
- 対策
  - サーバ OS 内部のメモリ上限を引き上げる
  - メモリ消費を抑える技術 syncache/syncookie

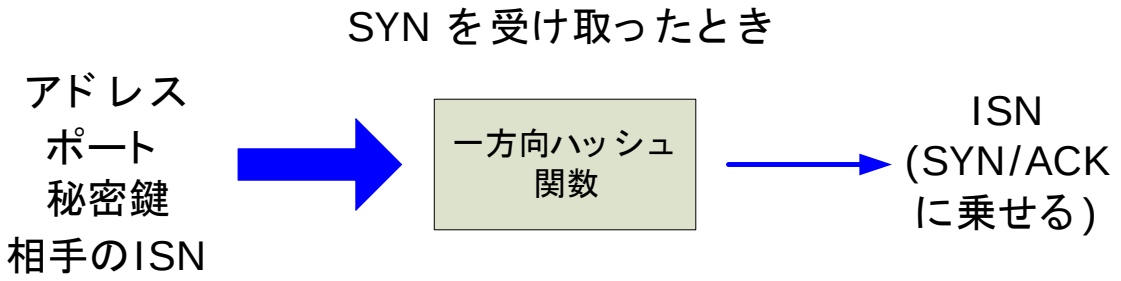
syncache

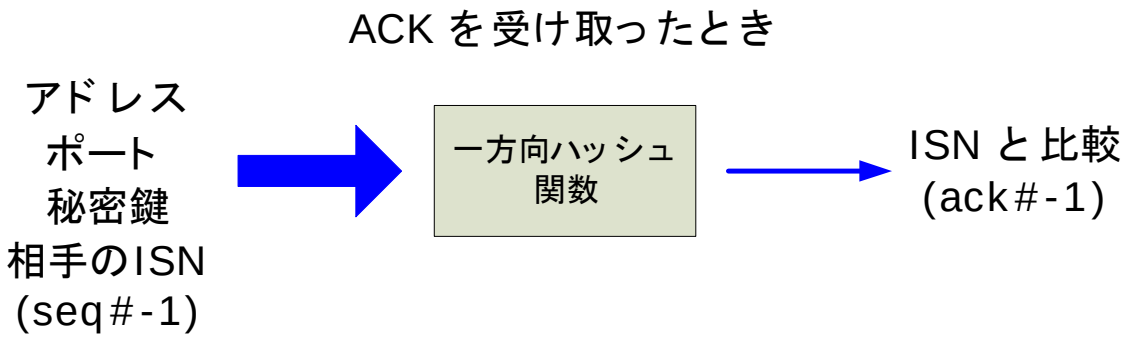
- 接続確立までメモリ消費を抑える技術
  - 接続確立まで不要な情報に対してはメモリを確保しない
  - 最小限の情報だけを通常データと分けて保持
  - ハンドシェイク未完了のエントリが区別/管理しやすい
    - 個数制限がかけやすい
    - 上限超過時に捨てるべきエントリを探しやすい



syncookie

- 接続確立までメモリ消費を 0 にする技術



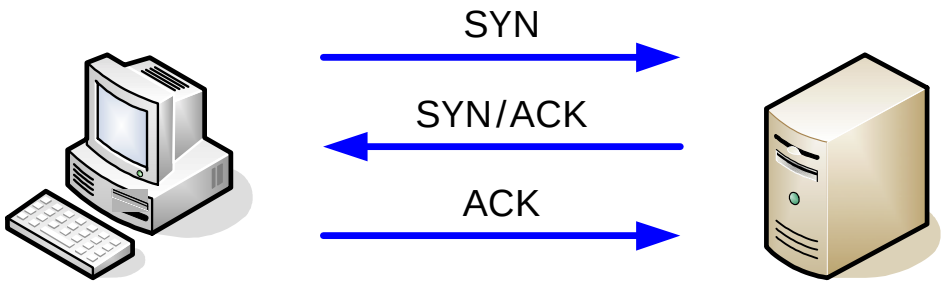


syncache/syncookie

- syncache と syncookie の併用 (FreeBSD の例)
  - まずは syncache を利用
  - メモリ上限に達したら、その後は syncookie を利用
- メモリ消費を大幅に抑えることで SYN flood への耐性を向上
- 3-way ハンドシェイクを完了させる型の SYN flood へは無力

TCP の脆弱性 ～ ISN 推測攻撃

- Initial Sequence Number (ISN)
  - 3-way ハンドシェイク時に sequence number を同期
  - アドレスを spoof していると SYN/ACK が受け取れない
    - 攻撃対象の sequence number がわからない
- この sequence number を推測できれば、spoof しつつ 3-way ハンドシェイクを完了できる

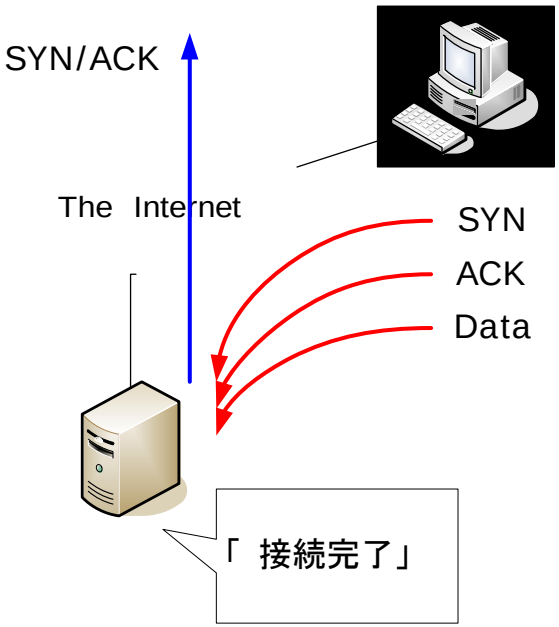


TCP の脆弱性 ～ ISN 推測攻撃

- ISN 推測が容易な実装がある
  - 一定時間ごとに ISN 値を一定分だけ増加
  - 接続要求(SYN)ごとに ISN 値を一定分だけ増加
- 仕様に定めはない

TCP の脆弱性 ～ ISN 推測攻撃

- 推測法の例
  1. 実際に接続してみて現在の ISN 値を得る
  2. spoof しつつ、一定分の増加を見込んでいくつかの sequence number で接続を試みる
- 試験環境内で spoof しつつ SMTP 接続し、メール送信に成功した例がある

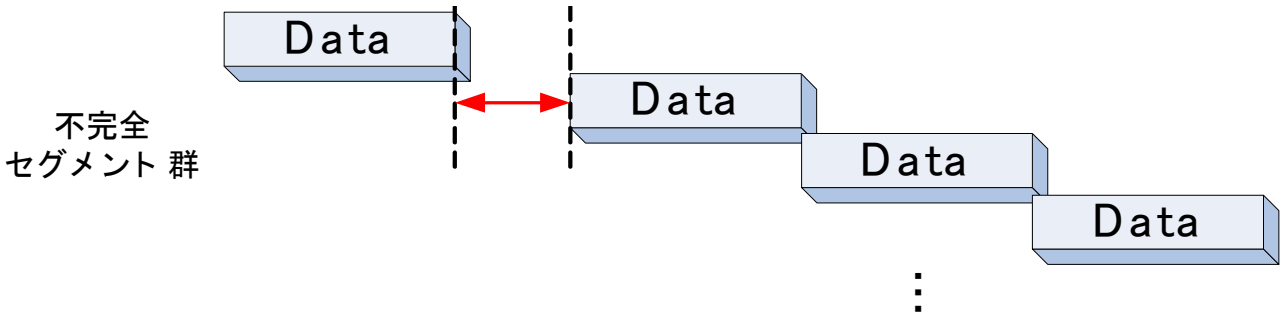


TCP の脆弱性 ～ ISN 推測攻撃

- 対策
  - ISN に乱数を利用
  - syncookie によるハッシュ値 ISN を利用
  - OS によって対応具合はまちまち

TCP の脆弱性 ～大量不完全セグメント攻撃

- 隙間があってリアセンブルできないセグメントを大量に送りつける
  - 仕様上はセグメントを保持しておく必要はない
  - 実際の OS では通信性能向上のために保持
  - mbuf が枯渇する可能性
    - ほとんど/まったく不応答になる
    - DoS



TCP の脆弱性 ～大量不完全セグメント攻撃

- 対策
  - OS 側で保持するセグメント数の上限を設ける
  - 上限を超過したらセグメントを破棄
    - もともと保持しなくても通信できる
    - 通常の通信では、まず上限を超えることはない
    - これまでの動作/性能を損わずに対策できる

TCP/IP プロトコルの脆弱性

- IP, ICMP, UDP, TCP の既知の脆弱性
- 低機能レイヤ: IP, ICMP, UDP
  - 単純なため、効果的な対策を加える余地が小さい
  - パケットフィルタ等、スタック外部の機能に頼ることが多い
- 高機能レイヤ: TCP
  - 複雑なため、仕様が想定していない異常時が多くあり、それが脆弱性の根源となりやすい
  - スタック外部からは状態が把握しにくい
  - スタック内部に対策が組み込まれることが多い