Distributed DDos Resistance

Mark Stapp, Cisco
CCNxCon, May 2015

Introduction

- Work-in-progress with Somaya Arianfar to explore ICN DDoS
- DDoS doesn't just go away in ICN
  - There will still be botnets
  - It will still be possible to send traffic at a target
    - it's not that hard to make up unique names...
- What ICN properties or protocols (such as denial messages) could help?

IP DDoS Characteristics

- Volumetric attack: generate traffic load from a large number of sources
  - Often using reflection (NTP, SSDP, etc.) as an amplifier
  - Especially effective where uRPF is not enabled
  - Note that the IP protocol isn't related to the target
- Attack traffic converges at the target, overwhelming legitimate traffic
  - Can be 100s of Gbps
- Perimeter defense (IDS) at the target network
  - DPI can detect patterns
  - But the traffic has already arrived
- The rest of the IP network just sees the 5-tuples (at most)
  - fq may make it even harder to detect a problem

Motivation

- ICN flavors of DDoS
  - Not convinced about 'satisfaction rate' at the edge
- Distribute network-layer information to improve network-wide response
  - Only the focus/target of an attack readily perceives it
  - DDoS traffic crosses administrative boundaries
  - Explicit signalling along attack paths that an attack has been detected
- Enable distributed/emergent mitigation?

Scenario

- Interests with routable target prefix, random 'names'
  - No specific reflection pattern in mind
  - Though we keep hearing proposals that will enable reflection
- Attacker clever enough to game 'satisfaction rate' at the edge
  - 5%? 10%? 15%?
  - Rate-limiting a prefix affects all traffic in the prefix

Denial-of-existence vs Denial-of-service

- Routers notice denial messages
  - Monitor 'background' level of denial
  - Identify prefix experiencing anomaly (standard deviation? order of magnitude?)
    - Threshold is local configuration
  - Use that to trigger first elevation
  - May use sat rate also
- Routers use manifests
  - Request manifest or denial manifest for suspect prefix
  - Begin sampling Interest names and test them
    - at a small rate initially
  - If samples test negative, raise the sampling rate
  - Don't have to test 100% at each hop; each additional hop contributes

Distributed

- Router processing takes place *outside* the target network
  - Greatest relief to the target
  - Routers could offload suspect prefixes for additional processing, dynamically
- No admin intervention is necessary
- No new signalling
  - Just using auth denial protocol messages
  - No standard (or trustworthy) signalling proto available in IP
- No explicit cooperation among routers
  - Each acts according to its own config
- Tunes to local conditions (load, compute power, etc)
- Legitimate traffic unaffected

Conclusions

- Some ICN characteristics that might support DDoS response
  - Changed sense of network layering that allows more explicit information to be visible to routers
  - Verifiable denial messaging
  - Stateful router processing
- Opportunity for individual routers to demonstrate useful dynamic behavior, without administrative action
- Enhanced uses for protocol features
  - Manifest/denial manifest

Backup

Manifest-based Denial

- Could manifests have a role in denial?
  - We'd like them to be distinguished, cacheable
- Name-hiding may be less important in ICN
- Subtle shift from "these are some names" to "these are the *only* names"
  - Implicitly provides denial along with the catalog
  - Add meta-data marker for terminal/childless names ("there are *no* children below this point")
  - Add meta-data marker for comprehensive list of children ("there are no *other* children below this point")

Manifest Example

```
<manifest>

<!-- Establish a ‘base name’ for a series of children. All
     of the children share the base, and any associated properties
     such as hash algorithm or signature parameters. -->

<basename>
  <name=“/example.com/data/docs/”/>
  <namehash=“SHA-256”/>

  <!-- Indicates that this block contains the complete set of names below
       the 'basename', possibly recursively. We'd expect that these child names
       would be self-certifying. -->
  <comprehensive='true' />

<object name=“doc1”>
  <segment id=0 hash=“0x...”/>
  <segment id=1 hash=“0x...”/>
  <segment id=2 hash=“0x...”/>
</object>

</basename>

</manifest>

<!-- Signature block for the manifest -->
<signature>
  [...]
</signature>
```

Dedicated Denial Manifest

- If name-hiding is important
- Combine the virtual namespace and manifest concepts
  - Create hashed version of the namespace
  - Construct a manifest using the hashed names
- Exposes little about the actual namespace
- Producer could return name of denial manifest in NAK message
- Cached denial manifest provides negative cache for an entire section of namespace
  - Rather than just a single gap in the 'simple' approach
  - Combine with "closest encloser" concept to support prefix-level denial
- Limited to fewer salt values

Denial Manifest Example

```
<manifest type='hashed'>

<!-- Manifest contains hashed names, not actual (routeable, etc.) names. Given
     the hash parameters, a client/cache can hash a target name and determine
     whether it falls into a gap in the hashed version of the namespace. -->

<hash-params>
<algorithm='SHA-256'/>
</hash-params>

<basename>
  <name="/example.com/data/docs/"/>
  <salt='xxx'/>

<!-- Hashed version of actual names, sorted, and/or references
     to child manifests -->

<name hashname="abcd..."/>
<name hashname="bbcc..."/>
<name hashname="ccdd..."/>

</basename>

</manifest>

<!-- Signature block for the manifest -->
<signature>
  [...]
</signature>
```