6.033: Security - Network Security
Lecture 24
Katrina LaCurts, lacurts@mit.edu

```
*********************************************************************
* Disclaimer: This is part of the security section in 6.033. Only   *
* use the information you learn in this portion of the class to     *
* secure your own systems, not to attack others.                    *
*********************************************************************
```

0. Today's Threat Model
   - Last week: adversary tried to observe or tamper with packets
   - Today: adversary is not just passively observing the network, but actively using it to attack users (more actively than the replay/reflection/on-path attacks we saw last time)
   - Some attacks today don't require adversary to observe packet contents; secure channels won't help

1. DDoS Attacks
   - Adversary's goal: bring down a service (e.g., take down the root DNS servers)
   - Strategy: congest the service. Make it spend time handling the adversary's requests so that it can't get to legitimate ones
   - DoS ("denial of service") attack
     - Adversary sends a bunch of traffic to the service (in many cases even invalid requests will work), queues fill up, packets dropped, etc.
   - DDoS ("distributed DoS") attack
     - Mount the attack from multiple machines
   - Can target any resource: bandwidth, routing systems, access to a database, etc.
   - Consequences of (D)DoS attacks
     - A server being down for a few hours might not seem like the end of the world. But..
     - Could be bank transactions
     - Could be DNS root servers (would bring Internet to a stand-still)
     - Could be on high-frequency trading machines, affect the stock market, etc.

2. Botnets
   - Can't we just toughen up our defenses? Add more bandwidth? How much traffic can one adversary generate?
   - Botnets: large (~100,000 machines) collection of compromised machines controlled by an attacker.
     - Make it very easy to mount DDoS attacks
     - Can be rented surprisingly cheaply
       - DO NOT DO THIS
   - How botnets work in five minutes
     - How do machines get compromised (and become part of the botnet)

```
    - Lots of ways.  Common way: user visits vulnerable website.
      Vulnerability is often a cross-site scripting attack.
      Example:
      - TrustedBlog.com has a box for users to enter comments on
        blogs.
      - Attacker embeds an executable script in his comment
      - When users browse, server sends comments to their browsers
        which execute the script, which sends the user's cookie to
        the attacker's site
    - XSS script to compromise a botnet machine causes user to
      download a "rootkit", which compromises the machine
      - see tomorrow's recitation
    - Bots contact command and control (C&C) servers which give
      them commands
  - How to combat botnets
    - Block IP addresses?  Ineffective.  Bots can change IP addresses
      rapidly.
    - Distribute systems so that DDoS attacks don't have a
      centralized component to bring down?  Not bad, but as we've
      seen, distribution => complexity

3. Network Intrusion Detection Systems (NIDS)
  - If we wanted to block IP addresses, how would we even figure out
    which IPs were part of the botnet?
  - Broader question: how do we detect network attacks?
  - Two approaches
    - Signature-based: Keep a database of known attack signatures and
      match traffic against the database.
      - Pros: Easy to understand the outcome, Accurate in detecting
        known attacks
      - Cons: Can't discover new attacks,Can only get the signature
        after the attack has already happened at least once
    - Anomaly-based: Match traffic against a model of normal traffic
      and flags abnormalities.
      - Pros: Can deal with new attacks
      - Cons: How do we model normal traffic?; Less accurate
        detection of known attacks
  - Many systems take a hybrid approach
    - Most also give users the ability to, once an attack is
      (passively) detected, do something to (actively) prevent it.

4. How to evade NIDS
  - Suppose we build a NIDS to scan traffic for a particular string
    ("root").  Seems easy.
  - Difficult because attacker can force confusing state on the NIDS
  - Another way to evade NIDS: mount an attack on the detection
    mechanism

5. Attacks that mimic legitimate traffic (and thus are even harder to
   detect)
```

- HTTP flooding
  - Attacker floods webserver with completely legitimate HTTP requests to download a large file or perform some computationally intensive database operation.
- TCP SYN floods
  - TCP connections start with a "handshake", which cause the server to keep some state about the connection until the client completes the handshake
  - Attacker can initiate many handshakes, exhaust state on the server
- Optimistic ACKs
  - Attacker starts TCP communication with victim, ACKs packets that it hasn't received yet
  - Victim sends more and more traffic to the attacker, saturating their own link
- DNS reflection/amplification
  - Bots locate DNS nameservers (even better if they are DNSSEC-enable)
  - Bots send DNS requests to these nameservers
    - Spoof sources to be the victim's IP address
    - If DNSSEC-enable, request the relevant info.  DNSSEC responses tend to be very large
  - Result: Large DNS responses that go to the victim's machine

6. Attacks on routers
   - Suppose adversary gains access to routers.  Could:
     - Overload the router CPU with lots of routing churns
     - Overload the routing table with too many routes
     - Hijack prefixes
       - Attacker gets an AS to announce that it originates a  prefix that it doesn't actually own.  Or to announce a more specific (and thus more-preferred) prefix.  Or to just lie that a shorter route exists.
       - Example: http://www.wired.com/2014/08/isp-bitcoin-theft/
       - Example: https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study
       - Example: https://www.theverge.com/2018/4/24/17275982/myetherwallet-hack-bgp-dns-hijacking-stolen-ethereum
       - Solution: secure BGP.  Similar mechanism as DNSSEC.  But, with authentication, creating advertisements (signing them) takes about 100 times as long as it does now.
         - Also need a lot of ASes to buy into this at once, otherwise it's not worth it

7. Moral of the story
   - Secure channels are great, but adversaries can still use the network to mount attacks
   - These attacks become devastating if they attack parts of the Internet's infrastructure (e.g., DNS, BGP)

- Proposals exist to secure the infrastructure (DNSSEC, Secure BGP), but there are problems
- All the more reason to build good, distributed, fault tolerant systems