Network Anti-Reconnaissance
Messing with Nmap Through Smoke and Mirrors

- AltF4
Anti-Reconnaissance

• Consider 3 main phases of a network attack:
  1) Gain Access
  2) Perform Reconnaissance
  3) Exploit Vulnerability

• Focusing on the second phase
  • Anti-Reconnaissance
  • Obscures the network
    – Obfuscate

• Not Intrusion Detection / Prevention

• Not Access Control
Reconnaissance: HowTo

• Find information to use in an exploit
  • Number of systems
    – ARP Sweep scan / ICMP Echo
  • Types (OS) of systems
    – OS detection scans
  • Open ports
    – TCP SYN / CONN (etc...) scans
  • Network Topology
    – Traceroute
  • Running Services
    – Service Detection Scans
Why Is Detecting Recon Hard?

• Signatures Fail
  • Identical at the packet level
    – ARP, TCP SYNs, ICMP, ...

• Speed
  • Being very slow can be stealthy
    – One packet per hour
  • Being very fast can be stealthy
    – Finish before anyone notices

• Already inside your network
  • Border security already bypassed (firewall)
Why Is Preventing Recon Hard?

- Metadata
  - Can't encrypt it
- Obfuscation
Constraining The Problem

- A Needle in a Haystack
  - Drown real nodes with realistic fake ones
  - Honeyd
- Two goals:
  - Obfuscates the network
    - Reconnaissance gets lots of bogus results
  - Identifies Reconnaissance
    - Traffic to decoys are presumptively hostile
Honeypots and Decoys

- Low Fidelity Honeypots
  - Not a real machine
  - Nor a “Virtual Machine“ as you know it
  - Can't be exploited like a VM can
  - Can be produced en masse
- Honeyd
  - Last update: 05/07/2007
  - Nmap new probes since then
    - nmap-os-db
  - github.com/datasoft/honeyd
Haystack

- Attacker gains access
  - Massive network
  - Most machines are fake
  - Can't tell the difference

- Reconnaissance becomes:
  - Ineffective
  - Cumbersome
  - Obvious
Classification

- High Fidelity Honeypots
  - Inspect log files
    - Manually
    - Maybe automated tools
- Signatures
  - IDS / Antivirus
  - Mostly fails
Machine Learning

- **K - Nearest Neighbors**
  - **N Statistical Features**
  - **Scalar Values**
    - Packet Timing
    - IPs contacted
    - Ports contacted
    - Haystack nodes contacted

- **Training Data**
  - Programmed into the system
    - Like a spam filter
  - Plot data points in N dimensional space
• Query Point
  • Search for the $k$ nearest neighbors
  • Majority vote
    – Distance metric

• **libann**
  • Approximate Nearest Neighbors
  • Introduces some error
  • Large performance gains
Features

- Haystack Autoconfig
  - Scans your network
  - Builds a Haystack from it
- Multiple UIs
  - WebUI, Qt, Terminal
- Import / Export Training Data
- Highly Multithreaded
- Free Software
Questions & Contact

- Email
  - altf4@phx2600.org

- Identi.ca
  - @altf4

- Twitter
  - @2600AltF4

- Diaspora
  - altf4@joindiaspora.com

- Development
  - github.com/DataSoft
  - IRC: OFTC #nova

- In Person
  - 1st Fridays, phx2600.org