Probabilistic Performance Analysis of Moving Target and Deception Reconnaissance Defenses

Michael Crouse, Bryan Prosser and Errin W. Fulp

October 12, 2015
Simplified Attack Process

- Most cyber attacks start with a reconnaissance phase
  - Gather intelligence about potential targets
  - Can require substantial time and effort for the attacker
- Consider defenses that occur at the reconnaissance phase
  - Attempt to render the attacker’s intelligence invalid
  - Potentially asymmetric given the attacker’s expense?
Reconnaissance Defenses

Two reconnaissance defense categories, movement and deception
Movement Defenses

- Movement used to change attack surface over time
  - After movement, reconnaissance information is invalid
  - Can be applied at different system levels

- Network address shuffling is one example of a moving target defense
  - Periodically remap network addresses
  - Reconnaissance information is nullified after a shuffle
  - Must properly manage legitimate traffic
Deception Defenses

- Masking, inventing, decoying, and mimicking used as a defense
  - False or misleading information provided to the attacker

- Honeypots are one example of a deception defense
  - A computer system designed to be a trap for attackers
  - Mimic a real system to entice the adversary into attacking
  - Slows the attacker and can collect important attacker information
Reconnaissance Defense Performance

- Recall, reconnaissance defenses are potentially asymmetric
  - Can potentially cost the attacker more than the defender
  - Empirical results suggest these defenses are successful

- What type of defense performance can we expect in other situations?
  - What is the probability of attacker success?
  - How should the defenses be deployed?
  - What networks are well suited for this defense?

Probabilistic models can perhaps provide greater insight into the performance of reconnaissance defenses
Network Model Assumptions

- There are $n$ addresses available (address space)
  - $v$ vulnerable computers, where $v \leq n$
  - $h$ honeypots within the network
- The attacker is aware of the address space ($n$ addresses)
  - Will attempt $k$ connections
  - Attacker wins if $m$ vulnerable computers located
  - Has probability $d$ of being deceived by a honeypot
  - Loses if honeypot(s) contacted (perhaps withstands some losses)
Urn Models

- Consider using urn models to estimate the attacker success
  - There are $v$ green marbles representing vulnerable computers
  - There are $h$ red marbles representing honeypots
  - Remaining are blue and represent empty or secure computers
- Drawing marbles from the urn represents scanning the network
  - Result of the draws (marble types) represent the attacker outcome
Undefended Model

- Consider the case where no defense is deployed
  - Probes $k$ unique addresses, no need to repeat an address
- Hypergeometric distribution models attacker outcome after $k$ probes

$$Pr(X_k = x) = \frac{\binom{v}{x} \binom{n-v}{k-x}}{\binom{n}{k}}$$

- Population consists of $v$ vulnerable and $n - v$ non-vulnerable
- Appropriate since draws are done “without replacement,” which represents the knowledge gained after each draw

- Attacker probability of success is given using the following

$$Pr(X_k \geq 1) = 1 - Pr(X_k = 0) = 1 - \frac{\binom{n-v}{k}}{\binom{n}{k}}$$
Honeypot Defense Model

- Consider the case where honeypots \( (h \geq 1) \) are deployed
  - Probes \( k \) unique address, but hopes to avoid honeypots
  - Let \( l \) represent the number of allowable losses

- Multivariate hypergeometric distribution models the attacker results

\[
Pr(X_k = x) = \frac{\binom{v}{x} \binom{h}{l} \binom{n-(v+h)}{k-(x+l)}}{\binom{n}{k}}
\]

- Population has \( h \) honeypots, \( v \) vulnerable, and non-vulnerable
- Draws are again done “without replacement,” which represents the knowledge gained after each draw
Attacker Success with Honeypot Defense

- Given the outcome probability, the probability of success is

\[
P_r(X_k \geq 1 \text{ and } l \leq L) = \sum_{l=0}^{L} \sum_{x=1}^{k-1} \frac{\binom{v}{x} \binom{h}{l} \binom{n-(v+h)}{k-(x+l)}}{\binom{n}{k}}
\]

- At least one vulnerable and no more than \( L \) losses (honeypots)

- Equations can be used to estimate attacker performance
Defense Analysis

- Foothold scenario
  - Attacker needs only one vulnerable computer in infrastructure
  - Foothold can then be leveraged to attack other computers

- Minimum-to-win scenario
  - Attacker wants to compromise a minimum number of computers
  - For example, information is distributed or the attacker is recruiting bots for a botnet
Consider a class-C network with 25% vulnerable computers

Honeypots reduces attacker success after a certain scan percentage
  - Larger reductions possible with slightly more honeypots
• Consider a class-C network with 25% vulnerable computers
  – Attacker must find at least 16 of the 64 vulnerable

• Honeypots still significantly reduces the attacker success
  – Overall min-to-win is a more difficult goal for the attacker
Effect of the Min-to-Win Objective

- Class-C network with 25% vulnerable computers and 5% honeypots
- Achieving the min-to-win can be increasingly difficult for the attacker
  - Hiding information across multiple computers is a good defense
Comparing Reconnaissance Performance

- Interested in comparing shuffle, honeypot, and combined defenses
  - Specifically the effect of the attacker wait time
- Assume a delay between reconnaissance and attack (*wait time*)

- If a shuffle occurs during the wait, then attacker likely fails
- This aspect (*wait time*) is not captured with shuffle models
- Instead, measure the effect of the different defenses empirically
Defense Simulations

• Discrete event simulator used to measure performance
  – Traffic traces (2008 SIGCOMM) replayed for connections
  – IP addresses remapped to a class-C network

• Defended network used shuffling, honeypots, or a combination
  – 10% of the network were vulnerable computers
  – For honeypot defense, 4% of the addresses were honeypots

• Random connections from the trace file were marked as reconnaissance
  – Attacker wait time was Poisson distribution with 10 minute average
  – If honeypots are used, attacker loses if a honeypot is contacted
  – If shuffling is used, attacker loses if a vulnerable computer is found (reconnaissance) but is not contacted after wait
  – Attacker wins if a vulnerable machine is contacted at the attack
Foothold Simulation Results

- Consider a 10% scan with different shuffle to wait ratios
  - If ratio is 2, then shuffle is 20 minutes (wait averages 10 minutes)
- Shuffling better than honeypot, if the shuffle interval less than wait
- Combined (shuffle and honeypot) always performs best
Min-to-Win Simulation Results

- Consider a 10% scan and vary the percentage required to win
  - Shuffle to wait ratio of 2 (20 minute shuffle interval)
- Honeypots better than shuffling if the minimum to win was low
- Combined (shuffle and honeypot) always performs best
Drop Probability Simulation Results

- Shuffling can disconnect illegitimate and legitimate traffic
  - Shorter intervals provide better defense, but higher drop probability
- Combined can provide good performance with longer shuffle intervals
Conclusions and Future Work

- Reconnaissance defenses attempt to invalidate attacker intelligence
  - Movement (shuffling) and deception (honeypot)
- Probabilistic models developed to estimate defense performance
  - Can provides defense analysis for various situations
- Models indicate only a few honeypots are necessary for a good defense
  - For class-C network, 5% (13) honeypots were very effective
  - Simulation results also indicate a combined defense is best
- Would like to develop a combined defense model
- Adding cost metrics would help determine best defense strategy
Turn back, you’ve gone too far
Shuffle Defense Model

• Consider a network address shuffling, with the following assumptions
  – Attacker scans then attacks with no wait
  – *Perfect shuffling* is used, therefore shuffle after every attack
  – There is no attacker knowledge gain after each attempt

• Binomial distribution models the attacker foothold outcome

\[ Pr(X_k = x) = \binom{v}{x} p^x (1 - p)^{k-x} \]

where \( p = \frac{v}{n} \)

– Population has \( v \) vulnerable and \( n - v \) non-vulnerable

– Draws are again done “*with replacement,*” which represents no knowledge gained after each draw
Attacker Success with Perfect Shuffle Defense

* Given the outcome probability, the probability of success is

\[ \Pr(X_k > 0) = 1 - \Pr(X_k = 0) = 1 - (1 - p)^k \]

* The previous equation is for foothold only
  - No penalty for multiples contacts of the same vulnerable computer

* For min-to-win, the population will change
  - If a green marble is drawn, it is replaced with a blue marble
  - This captures the notion that there is no benefit to contacting the same vulnerable computer more than once

* Probability of an attacker’s outcome follows a contagion equation

\[ \Pr(X_k = x) = \frac{(v)^x}{n^k} \cdot \sum_{j=0}^{x} (-1)^{x-j} \binom{x}{j} (n - v + j)^k \]
Shuffling Performance Analysis

- Similar probabilistic models exist for network address shuffling
  - Perfect shuffling modeled, remapping after every scan
  - Scan and attack occur simultaneously
- Shuffling does not perform well given these assumptions

![Graph showing attacker success probability versus percentage of address space scanned.]

- Shows the defense effect after an attack