

Behavioral Detection and Containment of Proximity Malware in Delay Tolerant Networks

Wei Peng, Feng Li, Xukai Zou, and Jie Wu

Proximity malware

Definition.

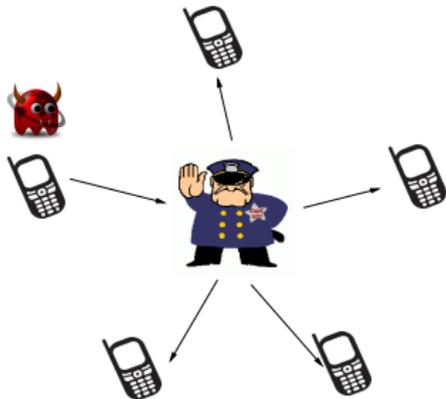
Proximity malware is
a malicious program
which propagates opportunistically...

... via Infrared, Bluetooth,
and more recently, Wi-Fi Direct.

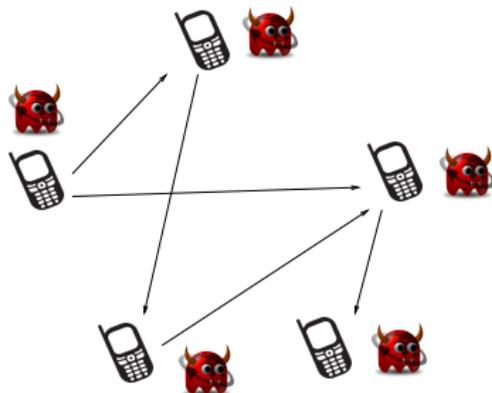
Proximity malware

Unique challenge.

Absence of a central gatekeeper (e.g., service provider) facilitates malware propagation.



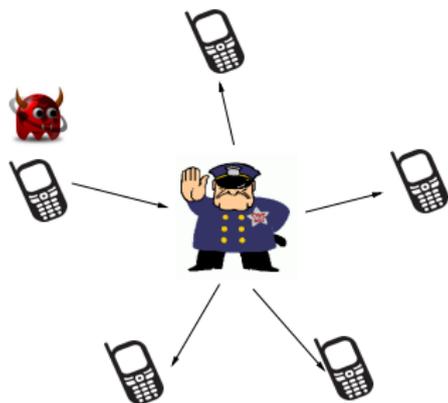
vs.



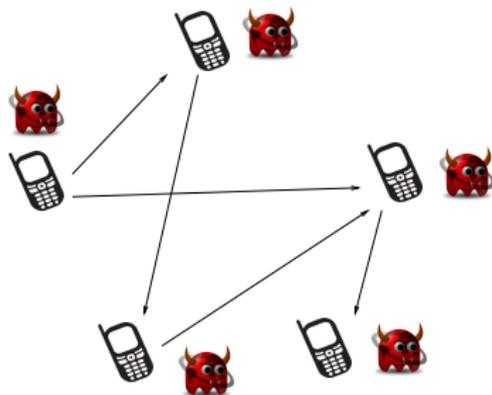
Proximity malware

Unique challenge.

Absence of a central gatekeeper (e.g., service provider) facilitates malware propagation.



vs.



Thus, **vulnerable** but **weak individuals** need to protect **themselves** from proximity malware.

Q: How to determine if a peer node is infected with malware?

Q: How to determine if a peer node is infected with malware?

A: By observing and assessing its behaviors

Behavioral characterization of proximity malware

Q: How to determine if a peer node is infected with malware?

A: By observing and assessing its behaviors in **multiple rounds**.



After smelling something burned



We have two choices

After smelling something burned



We have two choices



In the real life...

After smelling something burned



We have two choices



Cost?



Hyper-sensitivity leads to high false positive
while
hypo-sensitivity leads to high false negative.

To make the discussion concrete...

- DTN with n nodes.

To make the discussion concrete...

- DTN with n nodes.
- **Good** vs. **Evil**: nature of nodes based on malware infection.

To make the discussion concrete...

- DTN with n nodes.
- **Good** vs. **Evil**: nature of nodes based on malware infection.
- **Suspicious** vs. **Non-suspicious**: binary assessment after each encounter.

To make the discussion concrete...

- DTN with n nodes.
- **Good** vs. **Evil**: nature of nodes based on malware infection.
- **Suspicious** vs. **Non-suspicious**: binary assessment after each encounter.
 - **imperfect** good nodes may receive suspicious assessment (and vice versa) at times...

To make the discussion concrete...

- DTN with n nodes.
- **Good** vs. **Evil**: nature of nodes based on malware infection.
- **Suspicious** vs. **Non-suspicious**: binary assessment after each encounter.
 - **imperfect** good nodes may receive suspicious assessment (and vice versa) at times...
 - **functional** ... but most suspicious actions are correctly attributed to evil nodes.

Suspiciousness

... imperfect but functional assessment.

Node i has N (pair-wise) encounters with its neighbors and s_N of them are assessed as suspicious by the other party

Its **suspiciousness** S_i is defined as

$$S_i = \lim_{N \rightarrow \infty} \frac{s_N}{N}. \quad (1)$$

We draw a **fine line between good and evil**

$$L_e.$$

i is deemed good if

$$S_i \leq L_e$$

or evil if

$$S_i > L_e.$$

The question

How shall node i make the decision whether it shall cut off **future** communication with j based on **past** assessments $\mathcal{A} = (a_1, a_2, \dots, a_A)$?

Q: Where do the assessments \mathcal{A} come from?

A: Two models:

Household watch

i 's own assessments **only**.

Neighborhood watch

i 's own assessments **with its neighbors'**.

Household watch

Suspiciousness estimation and certainty.

Assume that the assessments are mutually independent.
To i , the probability that j has suspiciousness S_j given \mathcal{A} is

$$P(S_j|\mathcal{A}) \propto S_j^{s_{\mathcal{A}}}(1 - S_j)^{A - s_{\mathcal{A}}} \quad (2)$$

and the most likely suspiciousness is

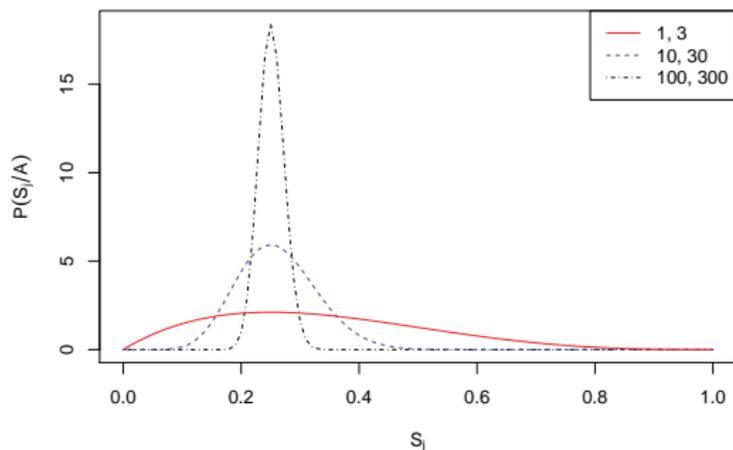
$$\arg \max_{S_j \in [0,1], \mathcal{A} \neq \emptyset} P(S_j|\mathcal{A}) = \frac{s_{\mathcal{A}}}{A}. \quad (3)$$

$s_{\mathcal{A}}$	The number of suspicious assessments in \mathcal{A} .
A	The number of assessments in \mathcal{A} .

Household watch

Suspiciousness estimation and certainty.

For different assessment sample sizes with
a quarter of suspicious assessments.



Though the most probable suspiciousness in all cases is 0.25,
the certainty in each case is different,
with 100 : 300 being the most certain one.

Household watch

Good or evil?

From i 's perspective, the probability that j is good is:

$$P_g(\mathcal{A}) = \int_0^{L_e} P(S_j|\mathcal{A}) dS_j, \quad (4)$$

and the probability that j is evil is:

$$P_e(\mathcal{A}) = 1 - P_g(\mathcal{A}) = \int_{L_e}^1 P(S_j|\mathcal{A}) dS_j. \quad (5)$$

Household watch

Good or evil?

Let $\mathcal{C} = (\int_0^1 S_j^{s_A} (1 - S_j)^{A-s_A})^{-1} dS_j$ be the (probability) normalization factor in Equation 2, we have:

$$P_g(\mathcal{A}) = \mathcal{C} \int_0^{L_e} S_j^{s_A} (1 - S_j)^{A-s_A} dS_j \quad (6)$$

and

$$P_e(\mathcal{A}) = \mathcal{C} \int_{L_e}^1 S_j^{s_A} (1 - S_j)^{A-s_A} dS_j. \quad (7)$$

Household watch

Good or evil?

$P_g(\mathcal{A}) \geq P_e(\mathcal{A})$ Evidence \mathcal{A} is **favorable** to j .
 $P_g(\mathcal{A}) < P_e(\mathcal{A})$ Evidence \mathcal{A} is **unfavorable** to j .

Household watch

Good or evil?

$P_g(\mathcal{A}) \geq P_e(\mathcal{A})$ Evidence \mathcal{A} is **favorable** to j .
 $P_g(\mathcal{A}) < P_e(\mathcal{A})$ Evidence \mathcal{A} is **unfavorable** to j .

Instead of making the cut- j -off decision right away
when $P_g(\mathcal{A}) < P_e(\mathcal{A})$,
 i **looks ahead** to confirm its decision.

Household watch

Look-ahead λ and λ -robustness.

Definition (Look-ahead λ)

The *look-ahead* λ is the number of steps i is willing to *look ahead before* making a cut-off decision.

Definition (λ -robustness)

At a particular point in i 's cut-off decision process against j (with assessment sequence $\mathcal{A} = (a_1, \dots, a_A)$), i 's decision of cutting j off is said to be *λ -step-ahead robust*, or simply *λ -robust*, if the estimated probability of j being good $P_g(\mathcal{A}')$ is still less than that of j being evil $P_e(\mathcal{A}')$ for $\mathcal{A}' = (\mathcal{A}, a_{A+1}, \dots, a_{A+\lambda})$, *even if the next λ assessments $(a_{A+1}, \dots, a_{A+\lambda})$ all turn out to be non-suspicious.*

Household watch

Look-ahead λ and λ -robustness.

Look-ahead λ is a **parameter** of the decision process rather than a **result** of it.

λ shows i 's willingness to expose to a higher infection risk in exchange for a (potentially) lower risk of cutting off a good neighbor.

In other words, λ reflects i 's **intrinsic** trade-off between staying connected (and hence receiving service) and keeping itself safe (from malware infection).

Household watch

Malware containment strategy.

i proceeds to cut j off
if the decision is λ -robust and
refrain from cutting off otherwise.

Neighborhood watch

Challenges.

- **Liars** Evil nodes whose purpose is to confuse other nodes by sharing false assessments.
- **Defectors** Nodes which change their nature due to malware infection.

Neighborhood watch

Naive evidence filtering.

- **Paranoia** Filter all and incorporate none. Degenerate to household watch with the twist of the defector problem.
- **Gullible** Filter none and incorporate all. Suffer from the liar problem.

Neighborhood watch

Naive evidence filtering.

- **Paranoia** Filter all and incorporate none. Degenerate to household watch with the twist of the defector problem.
- **Gullible** Filter none and incorporate all. Suffer from the liar problem.

Straightforward but not good enough!

Neighborhood watch

Evidence sharing.

Nodes share **direct, aggregate** assessments. Why?

- **Direct** No super-imposed trust relationship; one should not make trust decision for others.
- **Aggregate** Order of assessments does not matter in suspiciousness estimation shown in Equation (2).

Neighborhood watch

Defector problem: evidence aging window.

Only evidence within the **last T_E time window** is used in the cut-off decision process.

Evidence aging window T_E alleviates the defector problem.

Small enough to retire obsolete evidence.
Large enough for making the decision.

Neighborhood watch

Liar problem: dogmatism δ .

Definition (Dogmatism)

The **dogmatism** δ of a node i is the evidence filtering threshold in the neighborhood-watch model. i will use the evidence \mathcal{A}_k provided by its neighbor k within the evidence aging window T_E **only if** $|P_g(\mathcal{A} - \mathcal{A}_k) - P_g(\mathcal{A}_k)| \leq \delta$, in which \mathcal{A} is all of the evidence that i has (including its own assessments) within T_E .

Dogmatism δ alleviates
the liar problem.

Prevents the liars (the **minority** by assumption) to sway
 i 's view on the public opinion
of j 's suspiciousness S_j .

Neighborhood watch

Summary.

- Initialization.
 - Each node accumulates but does not use the evidence (aggregated assessment) provided by its neighbors.
 - During this phase, a node only uses its own assessments in making its cut-off decision.
- Post-initialization.
 - Each node starts to incorporate filtered evidence provided by its neighbors.
 - For a particular encounter, only if the evidence provided by the neighbor (within the evidence aging window T_E) passes the dogmatism test will the evidence provided *in this particular encounter* be used in making the cut-off decision.
 - Otherwise, all of the evidence provided by this neighbor within T_E will be ignored.

Contribution

- We give a general **behavioral characterization of proximity malware**, which allows for **functional but imperfect** assessments on malware presence.
- Under the behavioral malware characterization, and with a simple cut-off malware containment strategy, we formulate the malware detection process as a **decision problem**. We analyze the risk associated with the decision and design a simple yet effective malware containment strategy, *lookahead*, which is **distributed by nature** and reflects an individual node's **intrinsic trade-off between staying connected with other nodes and staying safe from malware**.
- We consider the benefits of **sharing assessments among directly connected nodes** and address the challenges derived from the DTN model in the presence of **liars** (i.e., malicious nodes sharing false assessments) and **defectors** (i.e., good nodes that have turned malicious due to malware infection).

Thank you!

Backup slides: verification.

Verification

Datasets.

	nodes	entries	time span	avg. interval
Haggle	41	89,836	12 days	12 secs
MIT reality	96	114,046	490 days	371 secs

Verification

Setup.

- $L_e = 0.5$.
- Randomly pick 10% of the nodes to be the evil nodes and assign them with suspiciousness greater than $L_e = 0.5$.
- The rest of the nodes are deemed as good nodes and are assigned suspiciousness less than $L_e = 0.5$.
- A random number is generated for each node in each encounter.
- A node receives a “suspicious” assessment if its random number is greater than its suspiciousness and receives a “non-suspicious” assessment otherwise.
- We choose an aging window of size of 20 minutes for Hagggle and 20 days for MIT reality.

Verification

Performance metrics.

	cut-off	no cut-off
evil neighbor	true positive	false negative
good neighbor	false positive	true negative

We sum up all of the corresponding decisions made by the **good** nodes and obtain four counts: TP (true positive), FN (false negative), TN (true negative), and FP (false positive). Then, the **detection rate** DR is defined as:

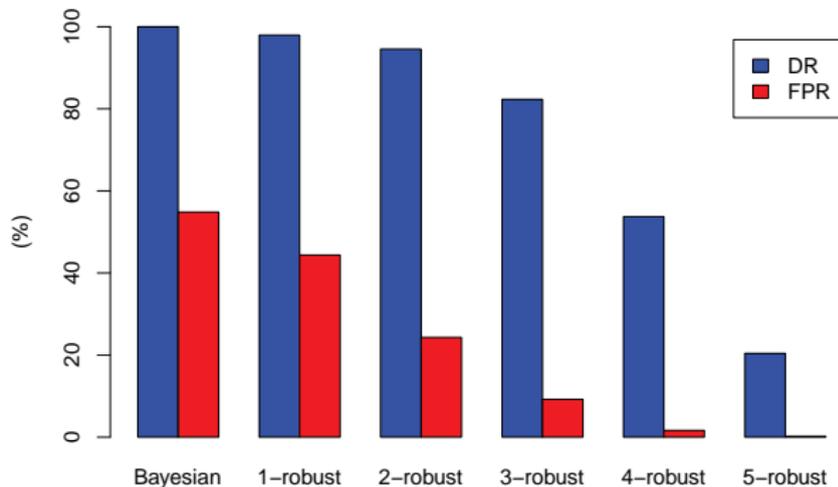
$$DR = \frac{TP}{TP + FN} \times 100\%,$$

and **false positive rate** FPR is defined as:

$$FPR = \frac{FP}{FP + TN} \times 100\%.$$

Verification

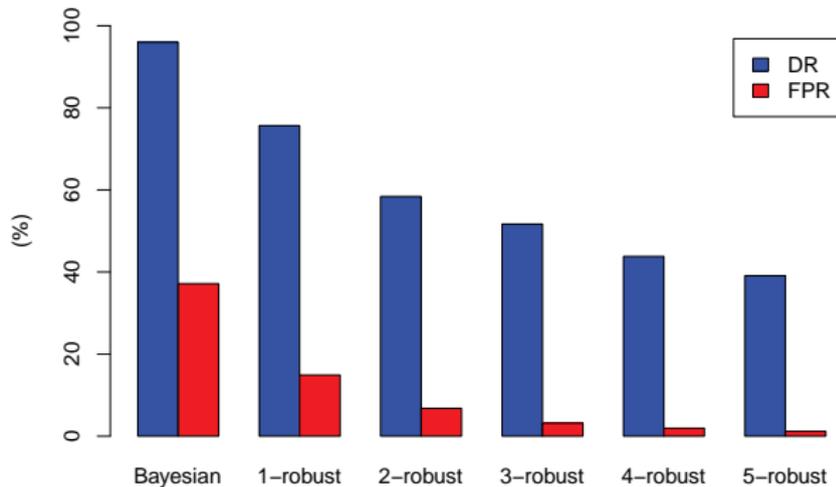
Result: look-ahead λ .



Bayesian decision with and without the look-ahead extension for Haggie. “Bayesian” shows the vanilla Bayesian decision; “ λ -robust” shows λ -robust decision.

Verification

Result: look-ahead λ .



Bayesian decision with and without the look-ahead extension for MIT reality. “Bayesian” shows the vanilla Bayesian decision; “ λ -robust” shows λ -robust decision.

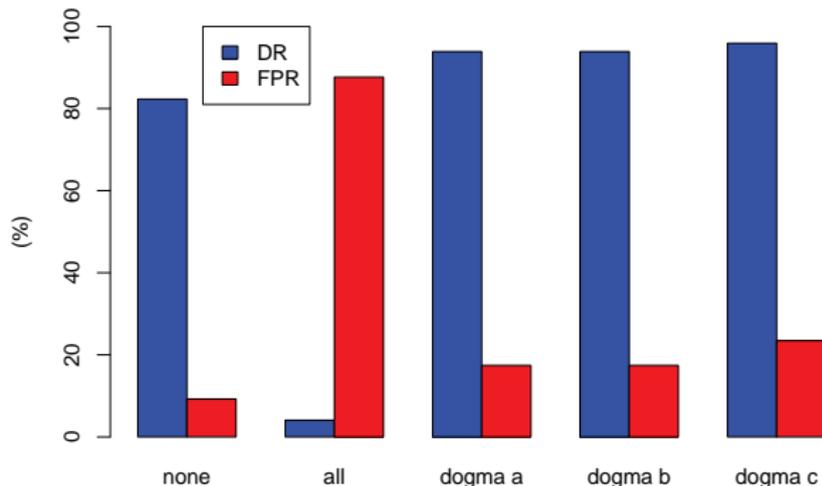
Verification

Result: look-ahead λ .

- In certain scenarios, trading a small decrease in detection rate for a large decrease in false positive rate is worthwhile.
- In those scenarios, the λ -robust decision process provides a simple yet effective method to stay connected while cutting off most connections with malware-infected nodes.

Verification

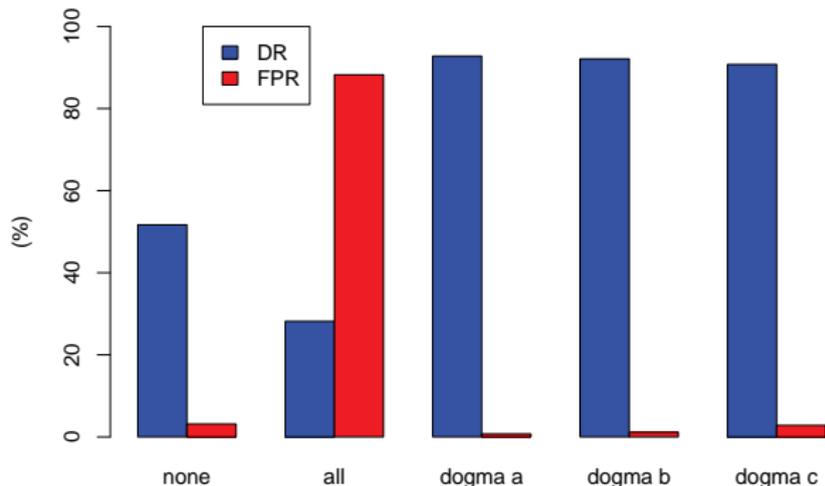
Result: dogmatism δ .



Effect of dogmatism δ on Haggie. Look-ahead is 3. “none” takes no indirect evidence; “all” takes all indirect evidence; “dogma” a, b, and c takes a dogmatism of 0.0001, 0.0010, and 0.0100, respectively.

Verification

Result: dogmatism δ .



Effect of dogmatism δ on MIT reality. Look-ahead is 3. “none” takes no indirect evidence; “all” takes all indirect evidence; “dogma” a, b, and c takes a dogmatism of 0.0001, 0.0010, and 0.0100, respectively.

Verification

Result: dogmatism δ .

- The “all” is rendered completely useless by taking all indirect evidence indiscriminately.
- In contrast, by filtering the evidence with the dogmatism test, the detection rate is increased (compared to “none”) with a modest increase in the false positive rate.
- The detection rate is almost doubled in MIT reality, which is in plain sight by comparing “none” and “dogma a”.