The Index Poisoning Attack in P2P File Sharing Systems

Shumanski, Andrei
Trigonakis, Vasileios

Agenda

- P2P File Sharing
- Systems under Evaluation
- Types of Attacks
- Solution
- Measurements & Results
- Conclusions

P2P File Sharing

- One of the most important applications in the Internet

↓

- Huge cost for the “copyright industry”

↓

- **Sharing systems under attack**

Terminology

- Title is a specific song or video
- A given title can have many different versions
- Each version has one identifier (hash of the version)
- Multiple copies of identical versions in the system
- Advertisements about the copies
- Keyword search is used

Agenda

- P2P File Sharing
- Systems under Evaluation
- Types of Attacks
- Solution
- Measurements & Results
- Conclusions

Systems under Evaluation

- **Overnet**:
  - DHT-based file sharing system
  - used in eDonkey2000 and eMule
- **FastTrack**:
  - two-tier unstructured file sharing system
  - used in Kazaa, Grokster and iMesh

Agenda

- P2P File Sharing
- Systems under Evaluation
- Types of Attacks
- Solution
- Measurements & Results
- Conclusions

Types of Attacks

- **Pollution attack**: making available corrupted content
  - Resource intensive attack
- **Index poisoning attack**: inserting massive numbers of bogus records into the index
  - Structured & unstructured systems
  - Requires less resources
- **Decoy attack**: either pollution or poisoning

The Index Poisoning Attack

- Typically, **no authentication** so it easy to advertise bogus information

- **Possible bogus information**:
  - non-existing, random ids (mostly used)
  - non-existing IPs
  - unavailable service port numbers

FastTrack & Index Poisoning Attack

- **Decentralized** & **unstructured** (two-tier)
- Two classes of nodes:
  - Ordinary-Nodes (ONs)
  - Super-Nodes (SNs)
- SN overlay, keeps the index

- **Attack by**:
  - inserting bogus records into the indexes of the SNs
  - TCP connection to a SN → publish bogus id/IP/port

Overnet & Index Poisoning Attack

- Based on **Kademlia**, all nodes equal
- UDP messages
- Two-step publishing:
  - Version ids
  - Keyword hashes
- **Attack by**:

i. defining the target keywords and hash them

ii. random id, not derived by some existing file, <u>**OR**</u> publish <key, value> and then <value, location>, where location is bogus

iii. periodically refresh this information

Agenda

- P2P File Sharing
- Systems under Evaluation
- Types of Attacks
- Solution
- Measurements & Results
- Conclusions

Solution – Methodology

- Downloading of files too expensive
- **Solution**:

i. **Harvesting**: collect the version ids and publisher node data & create a list of the advertised versions and a list of the distinct copies of each version. Done by:
   - **FastTrack**: a crawler
   - **Overnet**: inserting a node in the DHT with the target keywords hash as id

ii. Classify the versions (**clean, polluted, poisoned**)

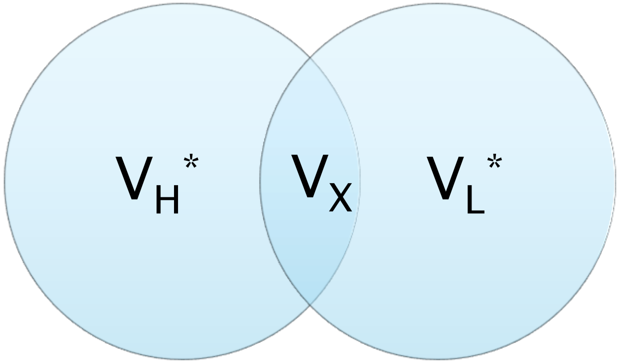
iii. Determine the pollution and poison levels for the versions and copies

Solution – Classifying the Users

- **Observation**: "*Among the users (with at least one version) the majority of users advertise a few versions* (**Light users**) *and a relatively small number of users advertise a large number of versions* (**Heavy users**)*.*"
- **U**: set of users advertised at least one version of a title
- **V_u^t**: for u ϵ U, the # of versions of title t from user u
- $V_u^{\max} = \max_{t \in T} V_u^t$: max # of versions for user u
- $m^{\max} = \frac{1}{|U|} \sum_{u \in U} V_u^{\max}$: the mean across all users
- K: constant so that: $\boxed{\text{u is } \textbf{Heavy user} \Leftrightarrow V_u^{\max} \geq K m^{\max}}$

Solution – Classifying the Versions

- **Heuristic**:
  - V → set of all the advertised versions
  - V_H → by heavy users
  - V_L → by light users
  - $V_X = V_H \cap V_L$ → **polluted versions**
  - $V_H^* = V_H - V_X$ → **poisoned versions**
  - $V_L^* = V_L - V_X$ → **clean versions**



- A normal user would advertise a small number of versions.

Poisoning & Pollution Levels

- **poisoning**:
  $|V_H^*| \; / \; |V|$
- **pollution**:
  $|V_X| \; / \; |V|$
- **clean**:
  $|V_L^*| \; / \; |V|$

Version Levels

- **poisoning**: $\dfrac{\sum_{u \in V_H^*} |C_u|}{\sum_{u \in V} |C_u|}$
- **pollution**: $\dfrac{\sum_{u \in V_X} |C_u|}{\sum_{u \in V} |C_u|}$
- **clean**: $\dfrac{\sum_{u \in V_L^*} |C_u|}{\sum_{u \in V} |C_u|}$

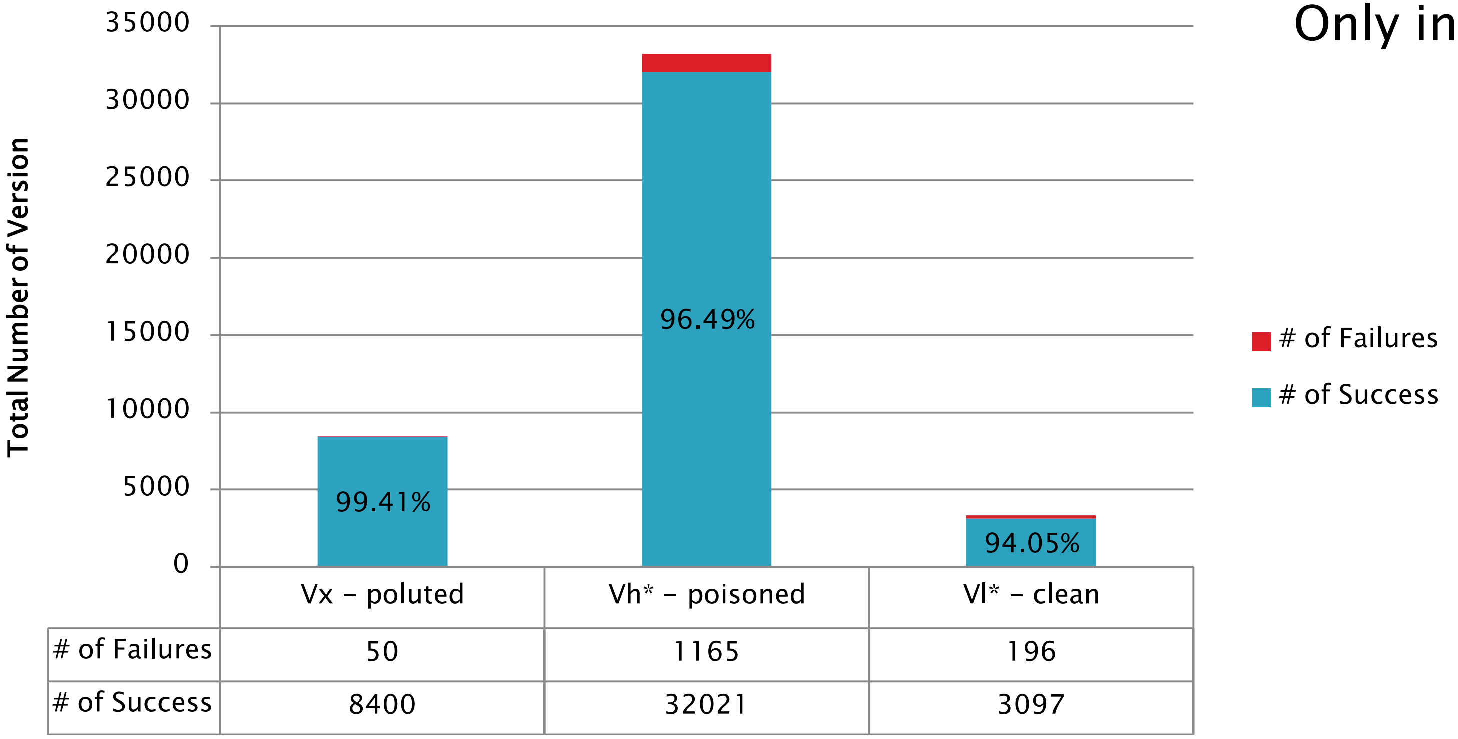
C_u is the set of copies for version u

Copy Levels

Agenda

- P2P File Sharing
- Systems under Evaluation
- Types of Attacks
- Solution
- Measurements & Results
- Conclusions

Evaluation of the Heuristic



	Vx – poluted	Vh* – poisoned	Vl* – clean
# of Failures	50	1165	196
# of Success	8400	32021	3097

Only in FastTrack

Overall, the scheme correctly classified more than **96%** of the versions.

Measurements & Results

- **FastTrack** (data set collected by the crawler in April 2005):

	# of IPs	# of users	# of copies	# of versions
Decoyer	624	8,683	1,183,622	443,102
Ordinary	82,015	117,673	347,939	167,103

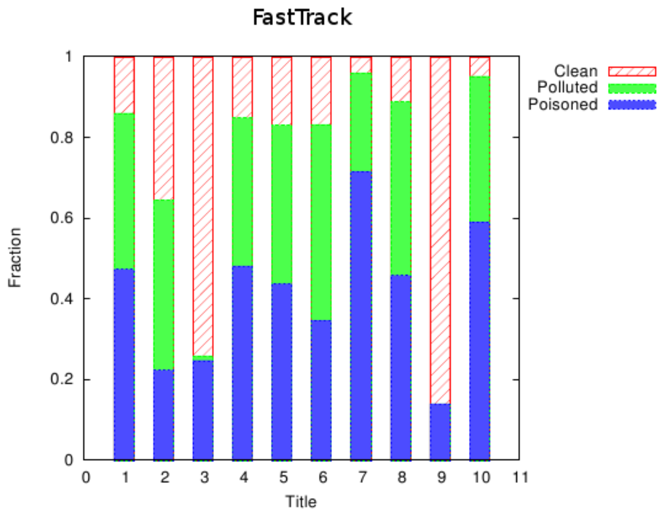
  - Decoyers are 7% of all users but provide 77% of all copies and 73% of all versions

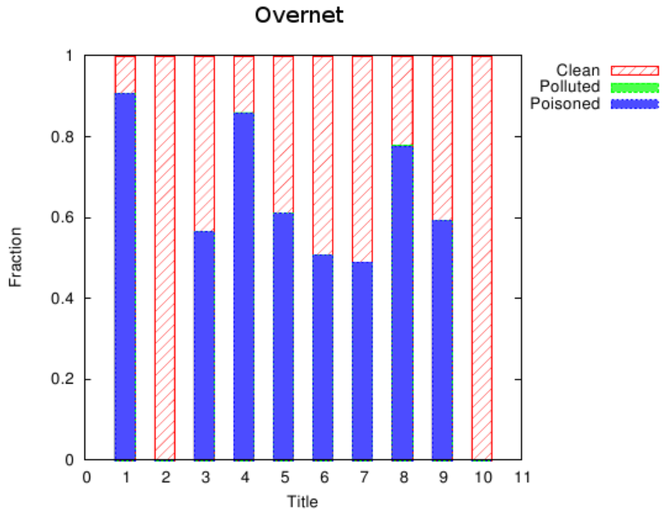
- **Overnet** (data set collected by the inserted nodes in June 2005):

	# of IPs	# of users	# of copies	# of versions
Decoyer	26	27	23,771	22,678
Ordinary	12,135	12,545	17,104	3,907

  - Decoyers are 0,2% of all users but provide 58% of all copies and 85% of all versions

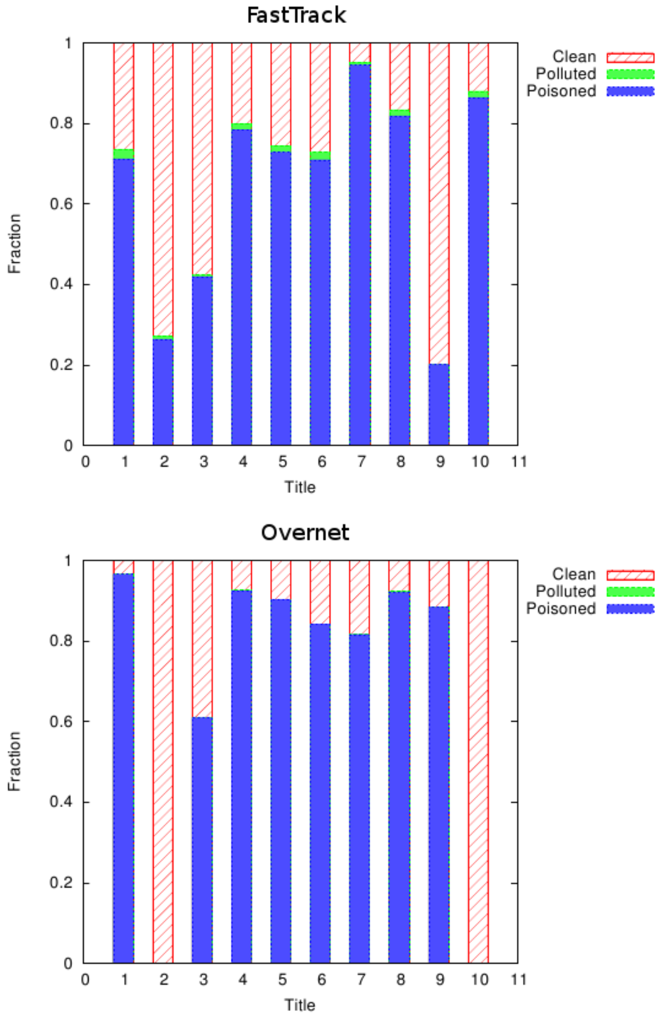
Mesurements & Results – Copies





- There are different companies and techniques
- Total decoy percentage is up to 95%
- Little pollution in Overnet

Mesurements & Results – Versions



- Majority of versions are poisoned
- Poisoning level up to 90%
- Differences in poison and pollution levels between versions and copies:
  - copies of the poisoned versions do not circulate
  - decoyers make many copies of the same polluted version

DHT Vulnerabilities to Poisoning

- **Node insertion attack**:
  - Not observed in FastTrack
  - Observed in Overnet – decoyers' nodes return random identifiers, prevent users from finding clean versions

- **Poisoning**: DHT vs. Unstructured
  - Small # of titles → DHT requires less resources
  - Increasing # of titles → eventually, DHT requires more resources

- **DDoS attack** by exploiting DHT
  - pointing one node

Defending against Poisoning Attack

- **Rating versions and advertisements** – forums

- **Rating sources**:
  - Reputation for range of IPs
  - Reputation based on number of copies per title
  - Nodes exchange reputation lists

Agenda

- P2P File Sharing
- Systems under Evaluation
- Types of Attacks
- Solution
- Measurements & Results
- Conclusions

Conclusions

- Both structured & unstructured overlays are vulnerable
- Heuristic to detect the polluted and poisoned versions/copies with a good approximation
- Defend by rating versions & sources
- DDoS attack possible in a DHT

References

- J. Liang, N. Naoumov, KW. Ross, *The index poisoning attack in p2p file sharing systems*, IEEE INFOCOM, 2006.

The end..

Thank you ☺