Automatically Discovering Surveillance Devices in the Cyberspace

Qiang Li, Xuan Feng, Haining Wang, Limin Sun
Jun 22, 2017
Outline

- Motivation
- System Design Considerations
- Fingerprint Generation
- Real-time Web Crawling
- Real-world Experiments
- Conclusion
A surveillance device is a type of digital video device typically deployed for monitoring the surrounding environment.
Surveillance device is everywhere
Surveillance device is everywhere

White House, USA

Tiananmen Square, China

outdoors

Indoors
Security and privacy

• Many surveillance devices are visible and accessible on the Internet.

• Surveillance devices have vulnerabilities that might be compromised.
Privacy concerns

• Monitoring public places
  – Street, market, office
• Hackers use vulnerabilities to watch some private cameras
  – Stranger hacks family's baby monitor and talks to child at night.
• God's Eye in Furious 7
  – Track someone
Security concerns

- Hackers turn security camera DVRs into worst Bitcoin miners
- Many surveillance devices, being exploited as parts of a “botnet”, could attack critical national infrastructures
  - October, 2016
  - Attack the Dyn Services
  - causing Internet service disruption across Europe and the United States
Security concerns

- Hackers Turn Security Camera DVRs Into Worst Bitcoin Miners
- Surveillance devices being exploited as parts of a “botnet”, attacking critical national infrastructures
  - October, 2016
  - attack the Dyn Services
  - causing Internet service disruption across Europe and the United States
Discovering surveillance devices

• A prerequisite
  – preventing from being compromised and exploited.
  – *help system administrators*
    • security auditing, detecting new kinds of vulnerabilities and intrusion
    • preserving device integrity on the Internet.

• Shed light on availability, reliability, and the distribution of these devices.

• Wiser decision for user and manufactures
  – plan a wiser decision based on the online activities of surveillance devices.
Discovering surveillance devices

• Traditionally discovery method uses manually marked key words
  – Shodan & Censys
  – arduous and error-prone process
  – hard to achieve completeness
  – hard to keep the discovery updated

• It is required to discover surveillance devices automatically and accurately
Observation

- User-friendly web interface
  - configure, access and manage a surveillance device conveniently
- Webpage of surveillance devices has two feature
  - relatively stable over time,
  - significant variability in different type of webpages

The diversity of webpages in the cyber-space.

Distinct appearances of surveillance webpages.
Technical challenges

• Many other embedded devices also have webpages
  – routers, network bridges, printers, and industrial control devices
  – commercial websites selling cameras would also affect our discovery

• Traditional web crawling is time-consuming
Our contributions

• We have proposed an automatic fingerprint generation approach for surveillance devices.
  – first work to use web appearance to identify online devices.

• Validation
  – implemented a prototype system and verified it in real-world experiments
  – results are promising (99% recall and 99% precision)

• Twice number of devices
  – four times over a five-month period in Amazon EC2
  – 1.6 million surveillance device discovery
Automatically fingerprint generating

Automatically generating fingerprints of surveillance devices
Pre-processing

- HTML parser (BeautifulSoup)
- Natural language processing (NLTK)
  - usually conjunctive words, delimiter-separated words, and letter-case separated words.
  - Stemming, remove the redundant text content
  - remove numbers, punctuations, and stopwords.
Device-related analysis

• Aim to seek common features of surveillance devices that are different from others.

• An iterative approach to extract common features for generating fingerprints of surveillance devices.
Classifier

• Classification Goal
  – building a classification model for determining whether a webpage belongs to a surveillance device.

• Based on the training data and the above feature space,
  – each webpage is transferred into a feature vector
    • each row is a feature vector as per page instance
    • each column is a value of the fields in the feature space.

• Machine learning algorithms for building the model to identify surveillance devices
Real-time web crawling

- Horizontal Scanner.
  - stateless connection, random permutation
- WEB Crawling
  - horizontal scanning to obtain the candidate
  - application-layer HTTP GET / request (root webpages)
  - HTTP Redirection
Real-world experiments

- Data Set (Ground Truth)
  - collecting 42,319 webpages
  - manually tag, cost about one month
  - 8,202 webpages of surveillance devices

- Training
  - 20,000 training datasets (3,847 surveillance devices)
  - remaining part, 2,2319 testing datasets (4,355 surveillance devices)
Classification performance

- The number of iterative processes
  - Number = 3
- For each iterative process, there are two parameters:
  - the feature selection algorithm and top N number
  - Chi2 test
  - top number N is set to 100
Classification performance

- The classification performance along with the number of the training set.
  - 500, 80%
  - 5,000, 96.5%
The overall accuracy for four classification models, SVM and KNN achieve best performance, choose SVM because it is capable of generating a maximum margin classifier with robustness.
Web crawling performance

- Hit rate = \( \frac{N_{\text{candidates}}}{N_{\text{total}}} \)
- Detection rate
  - the speed of discovering physical devices
- Hit rate drops down quickly, we speed up our detection rate
- 50 kpp/s gets best performance
Web crawling performance

The percentage of HTTP response status codes in the webpage crawling stage.

Time latency of online surveillance device discovery.
System overhead

• Training process
  – windows 10, 4vCPU, 8GB of memory
  – cost 10% of the CPU usage, 232MB of the memory usage.

• Online discovery
  – ubuntu14.04.2 LTS
  – 2 vCPU, 8GB of memory, 450Mbps bandwidth
  – CPU usage is 53%, average memory usage is 208Mbps
  – network bandwidth usage (out) is 50Mbps, 10% of network bandwidth

<table>
<thead>
<tr>
<th></th>
<th>CPU usage</th>
<th>Memory</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training process</td>
<td>10%</td>
<td>232.4MB</td>
<td>-</td>
</tr>
<tr>
<td>Online discovery</td>
<td>53%</td>
<td>208.9MB</td>
<td>50Mbps</td>
</tr>
</tbody>
</table>
Surveillance measurement campaigns

- Deployed our prototype system running on a Cloud server in Amazon EC2
- Four times from Sep 2015 to Jan 2016
- Scan 3.7 billion IP addresses
  - Blacklist, exclude 610 million IP addresses.
- Finding 1.6 million (1,602,142) surveillance devices,
  - nearly twice as many as existing search results.

<table>
<thead>
<tr>
<th>Begin Time</th>
<th>IP Space</th>
<th>Protocols</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-09-25</td>
<td>3.7 billion</td>
<td>HTTP</td>
<td>80, 8080</td>
</tr>
<tr>
<td>2015-12-22</td>
<td>0.36 billion</td>
<td>HTTP</td>
<td>80, 8080</td>
</tr>
<tr>
<td>2016-01-07</td>
<td>0.36 billion</td>
<td>HTTP</td>
<td>80, 8080</td>
</tr>
<tr>
<td>2016-01-05</td>
<td>3.7 billion</td>
<td>HTTP</td>
<td>80, 8080</td>
</tr>
</tbody>
</table>

Surveillance device discovery at the Internet scale.

<table>
<thead>
<tr>
<th>The number of surveillance devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shodan [3]</td>
</tr>
<tr>
<td>Censys [4]</td>
</tr>
<tr>
<td>Our system</td>
</tr>
</tbody>
</table>

Comparison with Shodan and Censys.
Conclusion

• We proposed a novel approach for automatic and accurate surveillance device searches.
• We proposed a new web crawling scheme
  – to obtain webpages of surveillance devices in a real-time and non-intrusive manner.
• We implemented a prototype of our approach and evaluated its performance through real-world experiments.
  – achieve 96% recall and 99% precision in surveillance device classification.
• Finding surveillance devices
  – twice as many as those using existing device search engines.
Thank you!  Q&A