Acquisitional Rule-based Engine for Discovering Internet-of-Things Devices

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

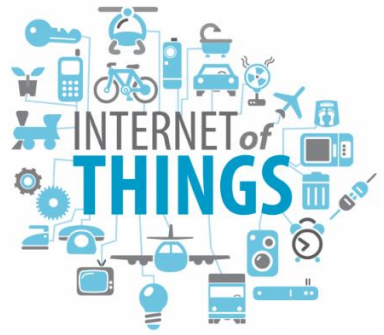
Xuan Feng, Qiang Li,
Haining Wang, Limin Sun
Jan 19, 2019

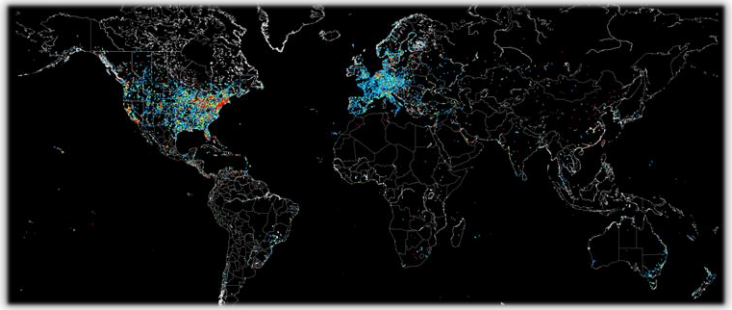
Outline

- Background and Motivation
- Rule Miner (ARE)
- Design and Implementation
- Evaluation
- ARE-based Applications
- Conclusion

Internet-of-Things (IoT) Devices

- **Various IoT devices connected to the Internet**
  - **cameras, routers, printers, TV set-top boxes,**
  - **industrial control systems and medical equipment.**
- **Estimated number – reported by Gartner**
  - **5.5 million new IoT devices every day**
  - **20 billion by 2020**
- **Meanwhile, these IoT devices also yield substantial security challenges**
  - **device vulnerabilities**
  - **mismanagement**
  - **misconfiguration**





Security Concerns

- Mirai botnet: IoT devices being compromised and exploited as parts of a “botnet”, attacking critical national infrastructures
  - October, 2016
  - attacking the Dyn Services
  - causing Internet service disruptions across Europe and the United States

- Hackers Turn IoT devices (DVRs) Into Worst Bitcoin Miners


COMPUTERWORLD
FROM IDG

Home > Security

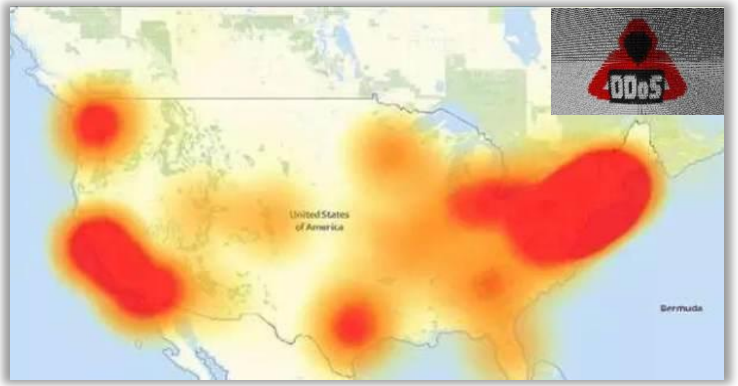
NEWS

DDoS attack on Dyn came from 100,000 infected devices

DNS service provider Dyn says Mirai-powered botnets were the primary source for Friday's disruption

By Michael Kan

U.S. Correspondent, IDG News Service | OCT 26, 2016 2:21 PM PT

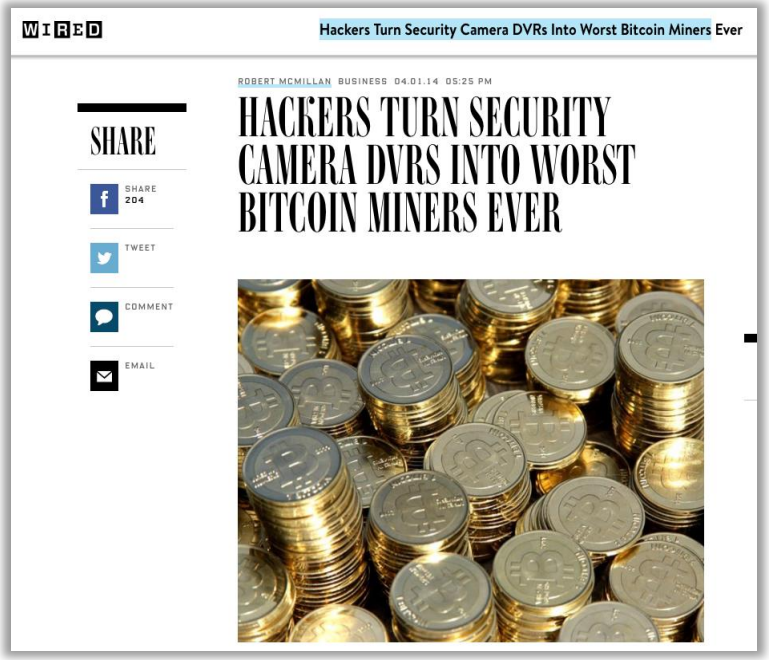


Map of areas most affected by Mirai attack

Security Concerns

- Mirai botnet: IoT devices being compromised and exploited as parts of a “botnet”, attacking critical national infrastructures
  - October, 2016
  - attacking the Dyn Services
  - causing Internet service disruptions across Europe and the United States

- Hackers turn compromised IoT devices (DVRs) into worst Bitcoin miners



Annotating IoT Devices

- There are two basic approaches to addressing security threats:
  - reactive defense
  - proactive prevention
    - more efficient than the reactive defense against large-scale security incidents

- To protect IoT devices in a proactive manner
  - a *prerequisite* step: discovering, cataloging, and annotating IoT devices.

Device Annotation

- The device annotation contains:
  - IoT device type (e.g., routers/camera),
  - vendor (e.g., Sony, CISCO),
  - product model (e.g., TV-IP302P).
- Fingerprinting-based Discovery.
  - high demand for training data and a large number of device models
- Banner-grabbing Discovery
  - examples: Nmap and Ztag
  - a manual fashion with technical knowledge
  - impossible for large-scale annotations
  - hard to keep the discovery updated

```
match http m|^HTTP/1\.1 400 Page not found\r\nServer:
IPCamera-Web\r\nDate: .* \d\d\d\d\r\nPragma:
no-cache\r\nCache-Control: no-cache\r\nContent-Type:
text/html\r\n\r\n<html><head><title>Document Error: Page
not found</title></head>\r\n\t\t<body><h2>Access Error:
Page not found</h2>\r\n\t\t<p>Bad request
type</p></body></html>\r\n\r\n| p/Tenvis IP camera admin
httpd/ d/webcam/
```

Regular expression used in Nmap

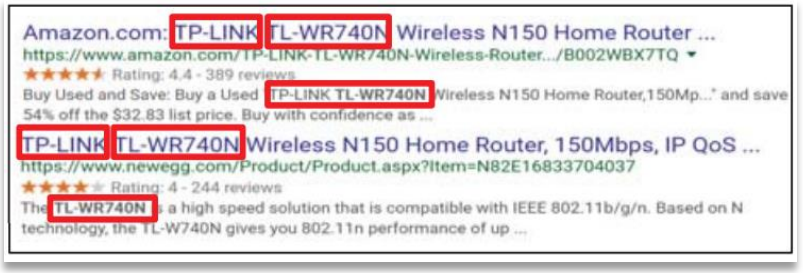
```
def process(self, obj, meta):
    cn = obj["certificate"]["parsed"]["subject"]["common_name"][0]
    if "Dell" in cn and "Printer" in cn:
        meta.global_metadata.device_type = Type.LASER_PRINTER
        meta.global_metadata.manufacturer = Manufacturer.DELL
        meta.tags.add("printer")
        meta.tags.add("embedded")
        if cn != "Dell Laser Printer":
            p = cn.split(" ")[1]
            meta.global_metadata.product = p
        return meta
```

Rules used in Ztag (Censys)

Key Observation

- Manufacturers usually hardcode the correlated information into IoT devices to distinguish their brands.
  - TL-WR740/TL-WR741ND in HTML file
- There are many websites describing device products such as product reviews.
  - Amazon and NEWEGG websites provide the device annotation descriptions.
- Our work is rule-based.
  - the automatic rule generation is mainly based on the *relationship* between the application data of IoT devices and the corresponding description websites.

Application layer data appears in IoT device.



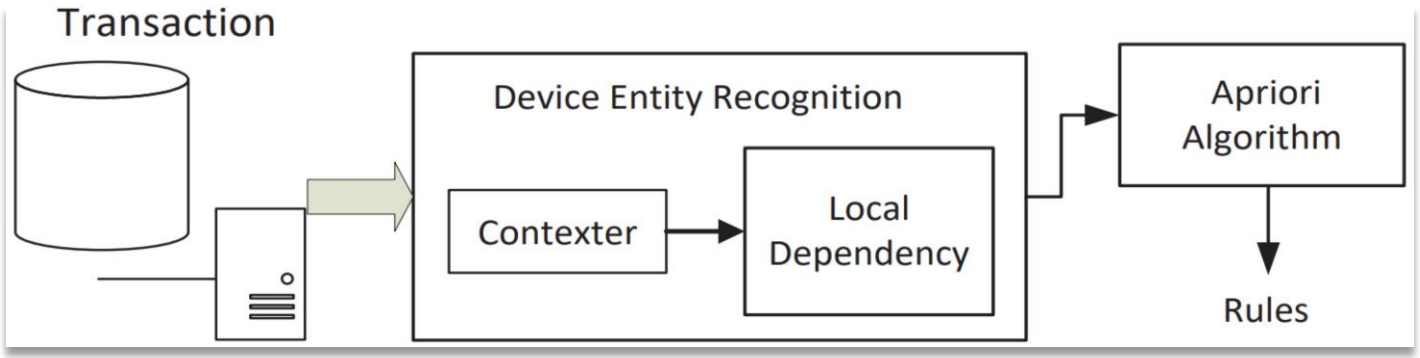
Relevant websites about this device in Google

Technical Challenges

- Two major challenges:
  - the application data is hardcoded by its manufacturer.
  - there are massive device annotations in the market.

- Notably, manufacturers would release new products and abandon outdated products.
  - manually enumerating every description webpage is impossible.

Rule Miner



Rule miner for automatic rule generation

- Transaction set
  - application-layer data and the relevant webpages
- Device entity recognition (DER)
  - contexter and local dependency
- Apriori algorithm
  - learn the relationship form Transactions

Transaction

- Transaction definition:
  - *a transaction is a pair of textual units, consisting of the application-layer data of an IoT device and the corresponding description of the IoT device from a webpage.*
- A rule is {A ⇒ B}.
  - the association between a few features (A) extracted from the application-layer data and the device annotation (B) extracted from relevant webpages

Device Entity Recognition (DER)

- DER is a combination of the corpus-based and rule-based.
  - corpus-based: device types and vendor names.
  - rule-based: use regular expressions to extract the product name entity.

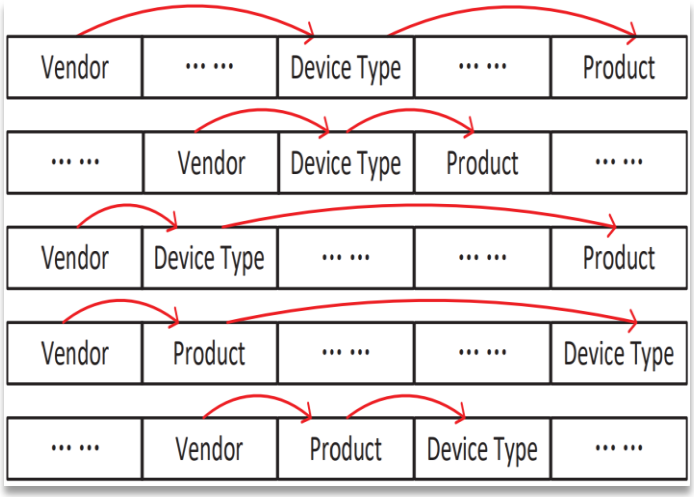
Entity	Context terms
Device Type	camera, ipcam, netcam, cam, dvr, router nvr, nvs, video server, video encoder, video recorder diskstation, rackstation, printer, copier, scanner switches, modem, switch, gateway, access point
Vendor	1,552 vendor names
Product	[A-Za-z]+[-]?[A-Za-z!]*[0-9]+[-]?[-]?[A-Za-z0-9]*^[0-9]2,4[A-Z]+

Context textual terms

Device Entity Recognition (DER)

- Poor performance :
  - high false positives in terms of device type and product name.
  - an irrelevant webpage may include keyword of device type such as “switch”.
  - a phrase that meets the requirement of regex for a product name.

- True IoT entities always have strong dependence upon one another.
  - (1) the vendor entity first appears, followed by the device-type entity, and finally the product entity;
  - (2) the vendor entity first appears, and the product entity appears second without any other object between the vendor entity, and the device-type entity follows



The local dependency of the device entity

Rule Generation

- Apriori algorithm

$$sup(A) = |\sum_{i}^{n} A \in t_i| / |T|$$

$$conf(A \Rightarrow B) = sup(A \bigcup B) / sup(A)$$

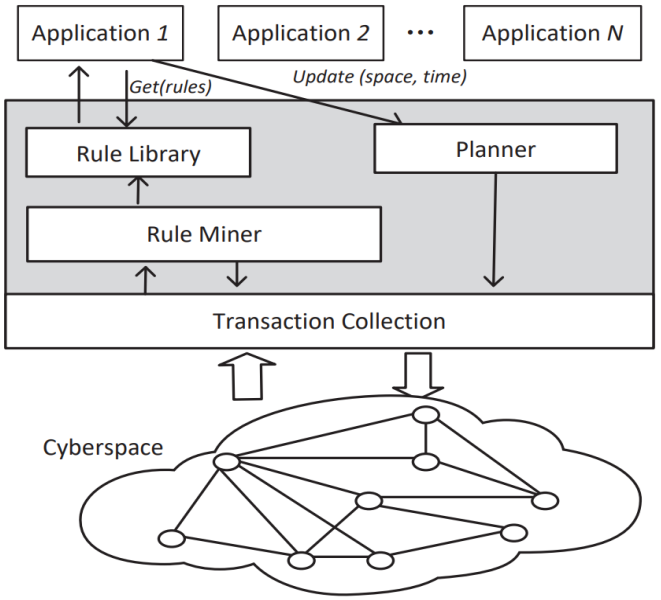
- Parameters
  - support is used to indicate the frequency of the variable (A) appearance
  - confidence is the frequency of the rules (A ⇒ B) under the condition in which the A appears
  - sup(A) = 0.1% and conf(A ⇒ B) = 50% work well.

Illustrating Rules
{ "Panasonic", "KX-HGW500-1.51" } ⇒ {IPCam, Panasonic,KX-HGW500}
{ "TL-WR1043ND", "Wireless","Gigabit", "ŏ0a9","Webserver" } ⇒ { Router, TP-Link, WR1043N }
{ "Welcome","ZyXEL", "P-660HN-51","micro_httpd"} ⇒ { Router, Zyxel, P-600HN }
{ "Juniper","Web", "Device","Manager", "SRX210HE","ŏ0a9" } ⇒ { Gateway, Juniper, SRX210 }
{ "Brother","HL-3170CDW", "seriesHL-3170CDW", "seriesPlease","debut/1.20" } ⇒ { Printer, Brother, HL-3170 }

A few example rules learned for IoT devices.

Design and Implementation

- Transaction collection
  - response data collection.
  - web crawler.
- Rule miner
- Rule library
  - store each rule {A ⇒ B}
- Planner.
  - update the rule library



Acquisitional Rule-based Engine (ARE) architecture for learning device rules.

Real-world Evaluation

- First dataset:
  - randomly choose 350 IoT devices from the Internet.
  - 4 different device types (NVR, NVS, router, and IPcamera) 64 different vendors, and 314 different products
- Second dataset:
  - 6.9 million IoT devices that our application collects on the Internet.
  - randomly sample 50 IoT devices iteratively for 20 times.
  - 1,000 devices across 10 device types and 77 vendors.

Real-world Evaluation

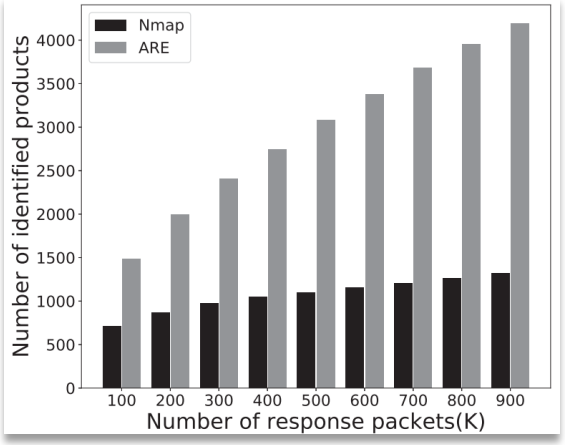
- Number of rules
  - generate 115,979 rules in one week.
  - in comparison with 6,514 from Nmap
  - 92.8% of rules - (device type, vendor, product).
  - 7.2% of rules just label device type and vendor.
  - about 30% of rules in Nmap with a fine-grained annotation.
- Precision of rules
  - first dataset: 95.7%
  - second dataset: 97.5%
- Coverage of rules
  - 94.9% coverage
  - given the same number of response packets, ARE achieves a larger coverage than Nmap

Category	Num	Percentage %
(device type, vendor, product)	107,627	92.8
(device type, vendor, null)	8,352	7.2

Rules generated by ARE.

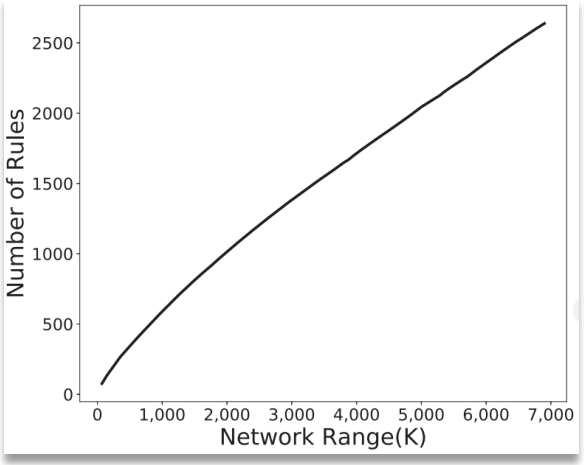
	Precision	Coverage
The first dataset	95.7%	94.9%
The second dataset	97.5%	—

Precision and coverage of rules on the dataset.



Real-world Evaluation

- Dynamic rule learning
  - the number of rules is increasing as ARE learns with the increase of network space.
- Overhead of ARE
  - Windows 10, 4vCPU, 16GB of memory, 64-bit OS
  - time cost of ARE for automatic rule generation is low in practice



Dynamic rule learning for ARE.

Stage	Latency (second)
Application layer data	0.5022
Response packet partition	0.0017
Web crawler	0.4236
Apriori algorithm	0.1166

Average time cost of one ARE rule generation.

ARE-based Applications

- Internet-wide measurement for IoT devices.
- Detecting compromised IoT devices.
- Detecting underlying vulnerable IoT devices.

Internet-wide Device Measurement

- Three application-layer datasets from Censys
  - HTTP, FTP, and Telnet.
- Deploying our collection module on the Amazon EC2
  - RTSP application-layer data.

- Using ARE, found 6.9 million IoT devices
  - 3.9M HTTP, 1.5M FTP, 1M Telnet, and 0.5 M RTSP.
- Discovery:
  - a large number of visible and reachable IoT devices on the Internet
  - the long-tail distribution is common for IoT devices (31% in Top 10)
  - many devices should not be visible or reachable from the external networks (camera/DVR).

Device Type	Number (%)	Vendor	Number (%)
Router	1,249,765 (18.3)	Mikrotik	641,982 (9.3)
NVR	785,810 (11.3)	Zte	352,498 (5.1)
DVR	644,813 (9.3)	Tp-link	325,751 (4.7)
Modem	466,286 (6.7)	Sonicwall	279,146 (4.0)
Camera	379,755 (5.5)	D-link	215,122 (3.1)
Switch	180,121 (2.6)	Dahua	153,627 (2.2)
Gateway	127,532 (1.8)	Hp	106,327 (1.5)
Diskstation	35,976 (0.5)	Asus	101,061 (1.5)

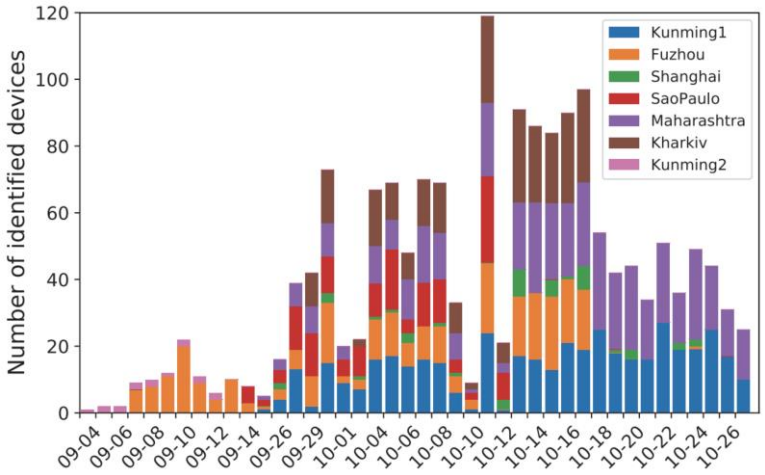
Automatic Internet-wide identification.

District	Number	Percentage (%)
United States	1,403,786	20.26
China	466,007	6.73
Brazil	442,781	6.39
India	297,446	4.29
Mexico	289,976	4.18
Taiwan	273,024	3.94
Republic of Korea	255,924	3.69
Russia	239,236	3.45
Egypt	204,237	2.95
Vietnam	199,415	2.88

Geographic distribution.

Compromised Device Detection

- Deploy honeypots as vantage points for monitoring traffic on the Internet.
- Annotating the captured IP addresses
  - a normal IoT device should never access honeypots.
  - an IoT device accesses our honeypots due to misconfigured or compromised.
- Honeypots
  - 4 countries, 7 cities
  - the duration is two months
- Discovery:
  - 50 compromised IoT devices every day.
  - In total, 2,000 compromised IoT devices among (12,928 IP addresses)
  - Device type: DVR, NAS and router
  - Also, some smart TV boxes exhibit malicious behaviors.



Compromised IoT device distribution.

Device Type	Num	(%)	Vendor	Num	(%)
DVR	1168	67.7	Hikvision	231	13.4
NAS	189	10.9	Dahua	216	12.5
Router	173	10.0	Qnap	189	10.9
Webcam	92	5.3	Mikrotik	81	4.7
Media device	83	4.8	TVT	79	4.5

Device type and vendor for compromised devices.

Vulnerable Device Analysis

- Finding underlying vulnerable devices
  - cross match the exposed IoT devices with the vulnerability information from NVD
- Discovery:
  - a large number of underlying vulnerable devices in the cyberspace
  - most vulnerabilities is about improper implementation
    - Path Traversal, Credentials Management, and Improper Access Control
    - Could be easily avoided if a developer pays more attention to security.

CWE ID	Weakness Summary	Number of IoT devices
200	Information Disclosure	573,656
22	Path Traversal	363,894
352	CSRF	348,031
264	Permission, Privileges, Access Control	345,175
255	Credentials Management	342,215
79	Cross-site Scripting	331,649
119	Buffer Overflow	149,984
399	Resource Management Errors	93,292
284	Improper Access Control	69,229
77	Command Injection	64727

Top 10 CWE of online IoT devices

Conclusion

- We propose the framework of ARE
  - automatically generate rules for IoT device recognition *without human effort* and *training data*.
- We implement a prototype of ARE and evaluate its effectiveness.
  - ARE generates a much larger number of rules within one week and achieves much more fine-grained IoT device discovery than existing tools.
- We apply ARE for three different IoT device discovery scenarios. Our main findings include
  - (1) a large number of IoT devices are accessible on the Internet
  - (2) thousands of overlooked IoT devices are compromised
  - (3) hundreds of thousands of IoT devices have underlying security vulnerabilities and are exposed to the public.

API and Dataset: http://are1.tech

Thank you! Q&A

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS