Faulds:
A Non-Parametric Iterative Classifier for Internet-Wide OS Fingerprinting

Zain Shamsi, Daren B.H. Cline, and Dmitri Loguinov

Internet Research Lab
Department of Computer Science and Engineering
Texas A&M University

Nov 1, 2017

Agenda

- Introduction
- OS Fingerprinting Techniques
- Iterative Classification with Faulds
- Internet Scan
- Conclusion

Introduction

- The Internet is exploding with different types of devices (e.g., phones, printers, cyberphysical, medical devices)
  - Builds unique attack vectors for hackers
  - Provides for interesting measurements for researchers
- One technique used for discovering these devices is **OS fingerprinting**
  - Determines the OS of a remote host
  - Can fingerprint specific firmware, which can reveal devices such as printers and webcams

Introduction

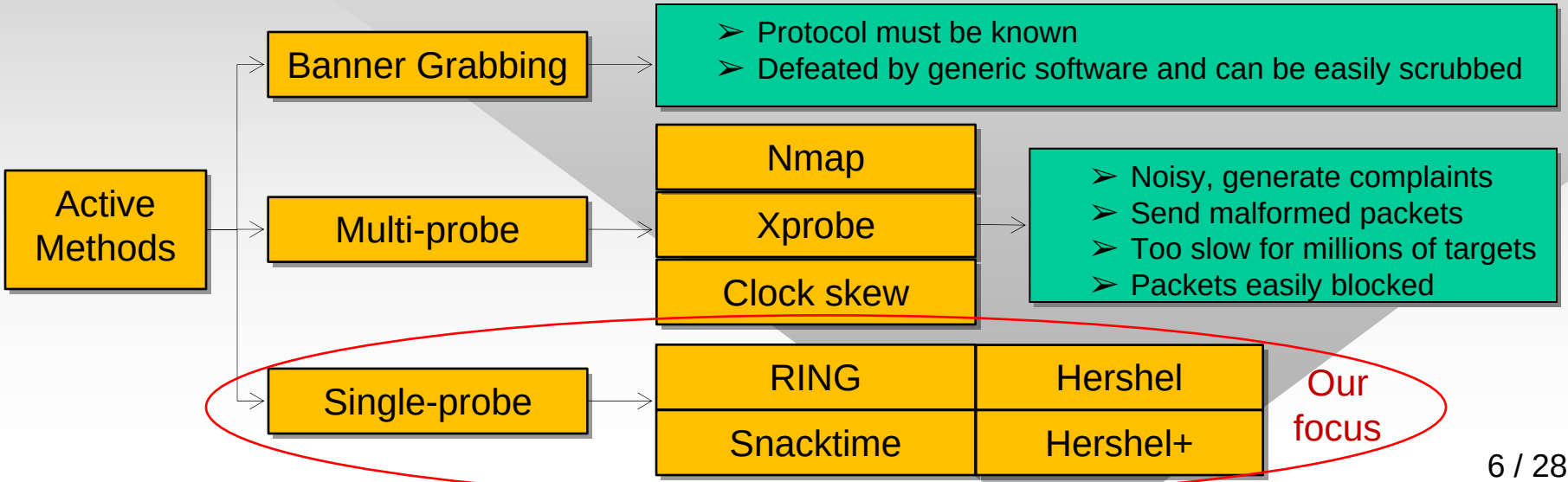
- OS fingerprinting is used widely in network security
  - Used by attackers as part of their reconnaissance/discovery
  - Used by administrators to survey their networks
  - Used by IDS/IPS to build better protections
  - Used by researchers/market analysts to measure networks
- Our focus in this work is improving the results of **large-scale** OS fingerprinting
  - We want to extract all the information we can
  - We do not want to increase our measurement footprint

Agenda

- Introduction
- Background: OS Fingerprinting
- Iterative Classification
- Internet Scan
- Conclusion

Background: Fingerprinting Methods

- For our purposes, we focus on **active** fingerprinting
  - Passive methods (e.g., p0f) require access to existing traffic
  - Active methods send crafted probes to elicit responses



Background: Single-Probe Fingerprinting

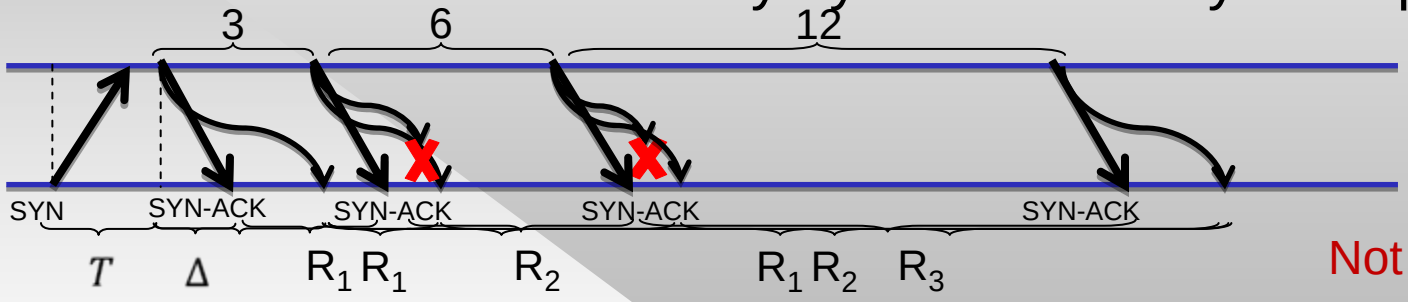
- Single-probe techniques use TCP probes and rely on two types of features for classification
  - Network features: Can be modified by network effects
  - User features: Can be modified by the end-user

OS	TCP WindowSize	IP TTL	IP DF	TCP Options	TCP MSS	SYN/ACK RTOs (in seconds)
Windows 7 / 2008	8192	128	1	MNWST	1460	3 6 12
HP Laserjet Series	8688	64	0	MNWNNSNNT	1456	2.9 6 12 23.9 29.9
Cisco IOS	536	255	0	M	536	1.9 4 8
	User Features					Network Features

M = MSS, S = SACK, T = Timestamp, W = Window Scale, N = NOP

Background: Network Features

- Network features are SYN/ACK retransmission timeouts (RTOs)
  - Can be affected drastically by network delays and packet loss



With delays	(2.8,6.4,12.1)					
1 packet lost	(9.2, 12.1)	(2.8, 18.5)	(2.8, 6.4)	(6.4, 12.1)		
2 packets lost	(21.3)	(6.4)	(18.5)	(9.2)	(12.1)	(2.8)
3 packets lost	empty					

Not just many possibilities, but also drastically different values!

Background: User Features

- User features can typically be modified in OS settings
  - Modification results in arbitrary value fluctuations
  - E.g., Receiver Window more likely to go from 8192 to 65535 than to 8193
- Treating all features as volatile, an observed sample can have multiple OS matches:

Fingerprint	TCP WindowSize	IP TTL	IP DF	TCP Options	TCP MSS	SYN/ACK RTOs (in seconds)
Observed	65535	128	1	MNW	1440	3.1 6.4
Windows 7	8192	128	1	MNWST	1460	3 6 12
HP Laserjet Series	65535	64	1	MNWNNTS	1456	2.9 6 12 23.9 29.9

Background: Hershel+ Review

- The most recent effort in single-probe fingerprinting is Hershel+ [Shamsi16]
- Let D be a database of known OSes, where each OS ω_i has a known feature vector, or fingerprint
- Suppose y is a random observation from some target
  - Hershel+ then determines the most probable ω_i that could have produced it: $\max_i p(\omega_i | y, \alpha_i, \theta)$

probability of ω_i given observation y

$\alpha_i = p(\omega_i)$ is the prior probability of ω_i

θ describes the noise model for delays (T, Δ), packet loss, and user tweaking

Background: Hershel+ Review

- Hershel+ classification uses a complex model relies on **prior** knowledge of (α, θ)
- Due to this, it has several assumptions:
  - α is treated as Uniform ($\alpha_i = \frac{1}{M}$ for all M OSes)
  - Forward delay T is treated as a fixed Erlang distribution f_T
  - One-way delay Δ is treated as a fixed Exponential dist. f_Δ
  - Packet loss probability q is fixed at 3.8%
  - Each user feature m has a fixed probability π_m to be tweaked

Background: Hershel+ Review

- These static assumptions do not allow flexibility in the network being studied
  - α = Uniform is most certainly not true in practice
  - Network conditions can change drastically
  - User feature modifications generally vary per device
    For example: A Windows host and Xerox printer will probably not be modified with equal probability
- The goal of our work is to solve these deficiencies and recover the true (α, θ)

Agenda

- Introduction
- Background: OS Fingerprinting
- Iterative Classification
- Internet Scan
- Conclusion

Iterative Classification

- Performance of Hershel+ depends on how well (α, θ) can be estimated a priori
  - An evolving Internet makes this extremely difficult
- We argue that (α, θ) should be the **output** of the classifier instead of the input
- We develop an iterative classifier called **Faulds** that produces this output as part of its classification
  - Named after Dr. Henry Faulds, credited with the first scientific study of human fingerprints in 1880

Iterative Classification with Faulds

- We derive several update equations for each parameter in (α, θ) to iteratively build the output
  - We prove that our updates fall under the Expectation-Maximization (EM) framework and follow its convergence properties

- For example, here is our result for estimating the prior probability for OS i, or α_i:

$$\alpha_i^{t+1} = \frac{1}{n}\sum_{j=1}^{n} p(\omega_i | y_j, \theta^t, \alpha^t)$$

At iteration $t+1$, α_i is the average probability with which observations match to signature ω_i

Conditioned on the previous estimates

Iterative Classification with Faulds

- Similar methodology follows for building distributions in the θ noise model (f_T, f_Δ, q, π)
  - Please see paper for update formulas and all derivations
- This treatment allows Faulds to customize its classification for the network being studied
  - Some networks may contain larger delays/loss (e.g., wireless)
- Faulds also builds per-device user feature models
  - Allows us to study user behavior on different classes of devices

Iterative Classification with Faulds

- We test Faulds using several simulation scenarios
  - Our plan is to expose our algorithm to various cases before using it in the wild
- Using a small database of 3 signatures, we simulated 2^{18} samples with different (α, θ)
  - Test database captures a mix of features

OS	TCP WindowSize	IP TTL	IP DF	TCP Options	TCP MSS	SYN/ACK RTOs
Linux 3.2	5792	64	1	MSTNW	1460	3 6
Windows 2003	16384	128	0	MNWNNTNNS	1380	3 6.5
Novell Netware	6144	128	1	MNWSNN	1460	1.4 3

Iterative Classification with Faulds

- We first focus on simulating different network conditions

Simulation Scenario 1: (Network only)	Hershel+		Faulds	
Real α = (0.9, 0.05, 0.05) Network parameters follow assumptions f_T = Erlang(2), f_Δ = Exponential Packet loss rate = 3.8%	Acc.	Recovered α	Acc.	Recovered α
	67%	(0.59, 0.35, 0.06)	95%	(0.89, 0.06, 0.05)
	Recovery of α drives Faulds improvement			

Simulation Scenario 2: (Network only)	Hershel+		Faulds	
Real α = (0.9, 0.05, 0.05) f_T = Pareto, f_Δ = Pareto Non-uniform packet loss 3-packet drop rate = 10% 4-packet drop rate = 66%	Acc.	Recovered α	Acc.	Recovered α
	45%	(0.34, 0.47, 0.19)	97%	(0.90, 0.05, 0.05)
	Faulds correctly recovers α, f_T, f_Δ, q			

Iterative Classification with Faulds

- We now switch to modifications to user features

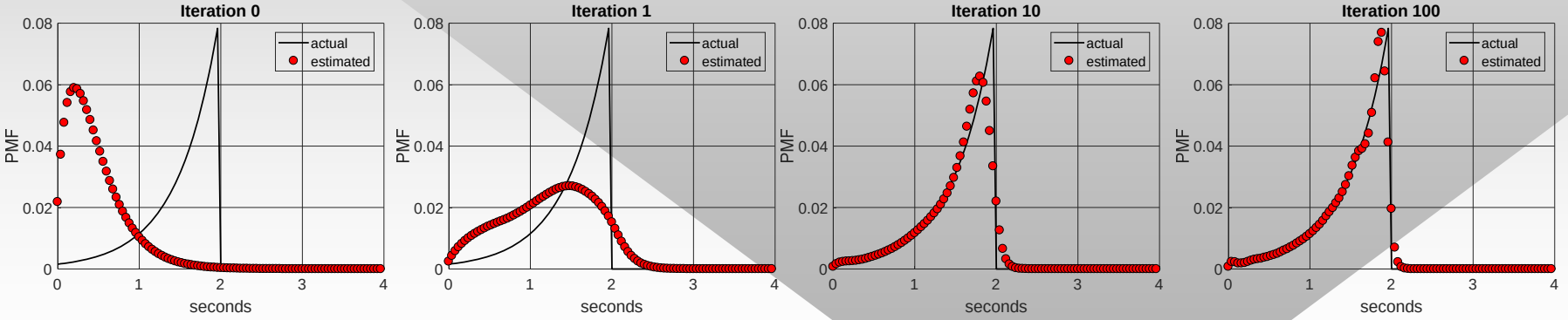
Simulation Scenario 3: (User only)	Hershel+		Faulds	
Real α = (0.7, 0.2, 0.1) OS patches cause a change in each user feature with probability 80% ($\pi_m = 0.8$ for all m)	Acc.	Recovered α	Acc.	Recovered α
	31%	(0.09, 0.59, 0.32)	100%	(0.7, 0.2, 0.1)
	Faulds learns new post-patch values to get to 100%			

Simulation Scenario 4: (User only)	Hershel+		Faulds	
Real α = (0.7, 0.2, 0.1) Users deploy scrubbers that change user features non-uniformly $\pi = (1.0, 1.0, 0.9, 0.8, 1.0)$	Acc.	Recovered α	Acc.	Recovered α
	29%	(0.06, 0.43, 0.51)	91%	(0.7, 0.2, 0.1)
	Hershel+ is tricked, Faulds learns the true π vectors			

  - Please see paper for more scenarios

Iterative Classification with Faulds

- We also test Faulds on the full Hershel+ database which contains 420 OS signatures
  - Simulate counter-intuitive examples – such as set f_Δ to be a reverse-exponential distribution (larger delays more likely)



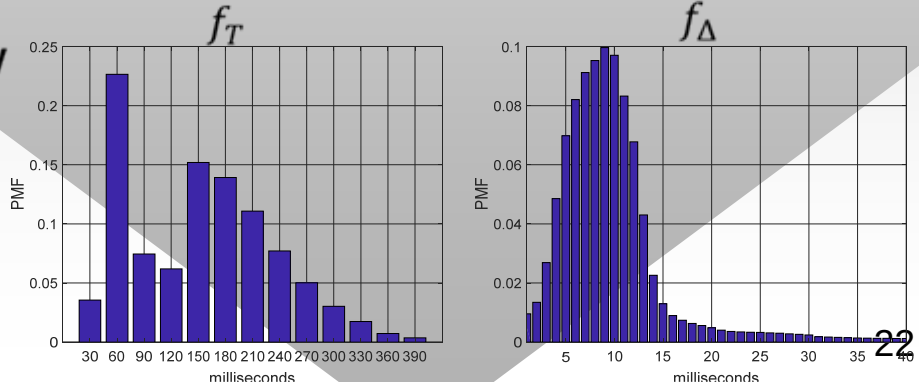
- Showed Faulds can also recover complex distributions

Agenda

- Introduction
- Background
- Iterative Classification
- Internet Scan
- Conclusion

Internet Scan

- In December 2016, we conducted a SYN scan of all BGP-reachable IPv4 on port 80
  - About 2.9 billion IPs contacted, received 67.6M responses
- We unleashed Faulds on this dataset for 100 iterations using the Hershel+ database
  - Our final f_T and f_Δ show average T = 148ms and average Δ = 15ms



Internet Scan

- During classification, Faulds consolidates several signatures by learning how much they are tweaked
  - Top 10 classified OSes:

Iteration 1

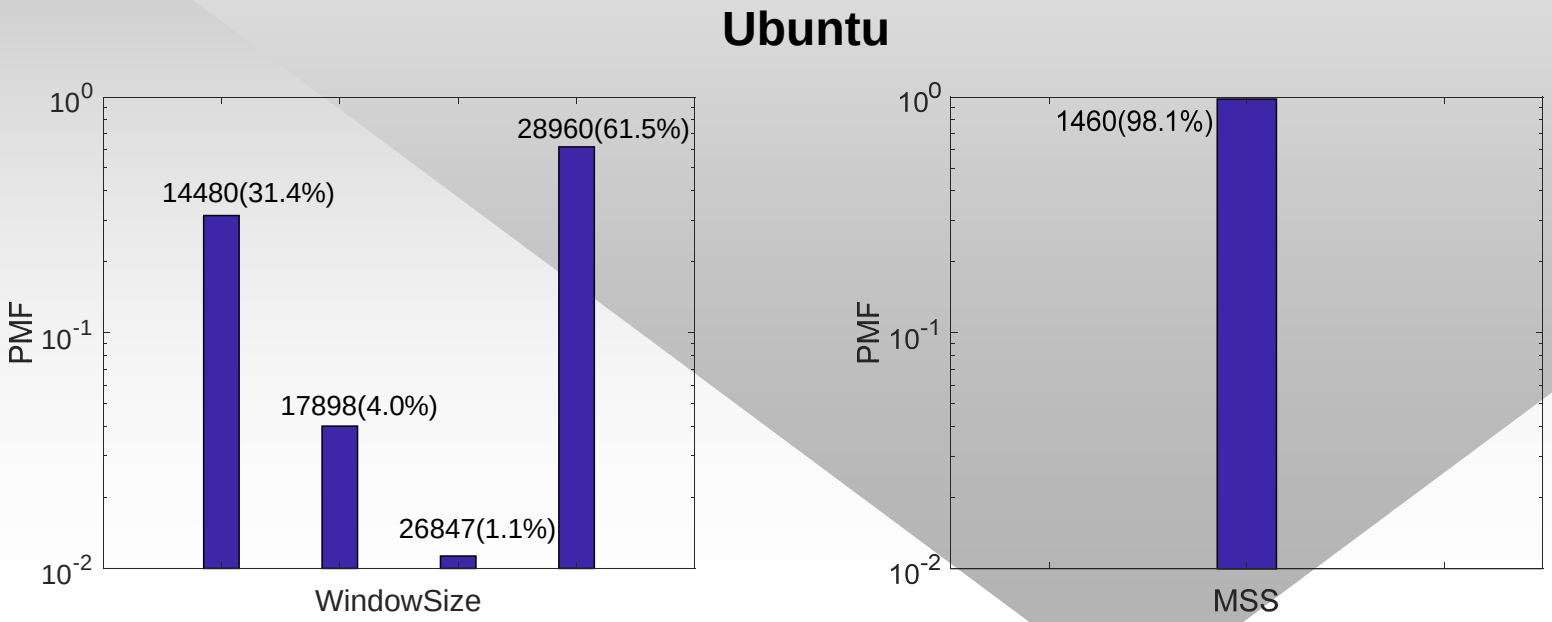
OS	α_i	Count
Ubuntu/Redhat/CentOS	22.4%	14.01 M
Debian/SUSE	11.1%	8.89 M
Embedded Linux	8.2%	6.32 M
Windows 7 / 10 / 2008	4.7%	2.94 M
Redhat/SUSE	3.7%	2.40 M
Schneider/APC Embedded	2.2%	1.58 M
Windows XP / 2003	2.1%	1.31 M
Redhat/CentOS	1.8%	1.24 M
Embedded Linux	1.5%	1.04 M
Windows 2008 R2 / 2012	1.3%	907 K

Iteration 100

OS	α_i	Count
Ubuntu/Redhat/CentOS	33.4%	21.36 M
Embedded Linux	10.3%	6.47 M
Windows 7 / 10 / 2008	5.6%	3.66 M
Schneider/APC Embedded	5.5%	3.63 M
Redhat/SUSE	3.1%	2.00 M
Windows XP / 2003	1.8%	1.24 M
Redhat/CentOS	1.6%	1.04 M
Dell Laser / Xerox Printers	1.5%	976 M
Windows 2008 R2 / 2012	1.4 %	837 M
Cisco Embedded	1.3%	824 M

Internet Scan

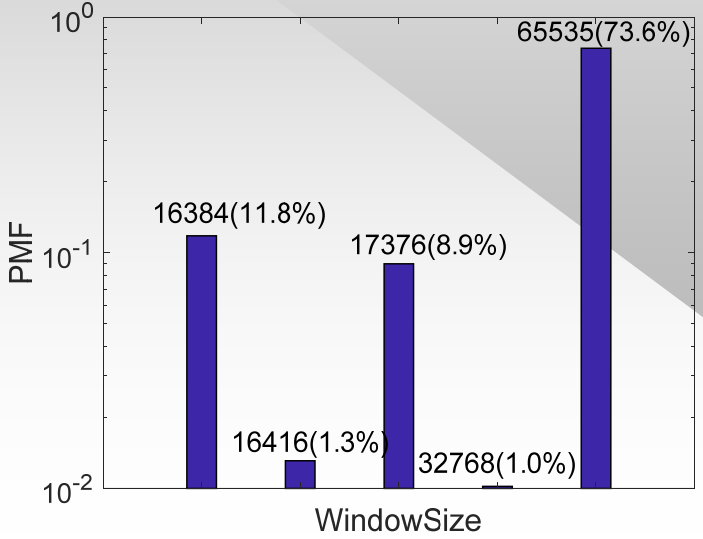
- Faulds outputs interesting details about end-user behavior in modifying each device in the database

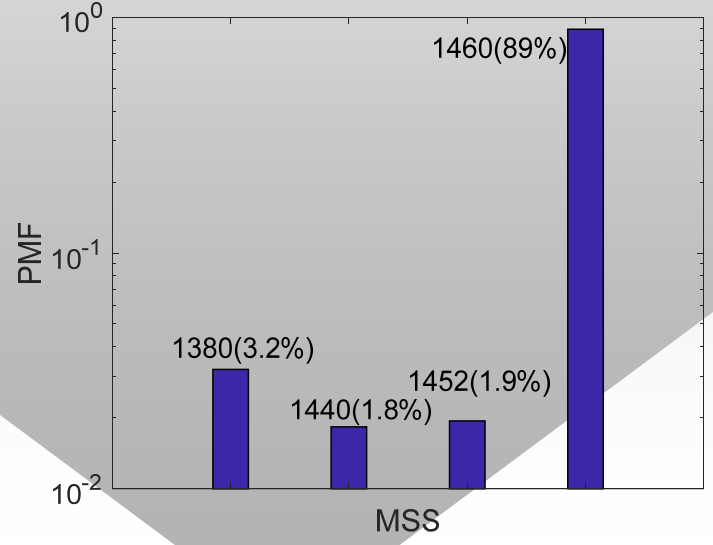


Internet Scan

- Faulds outputs interesting details about end-user behavior in modifying each device in the database

Mac OSX

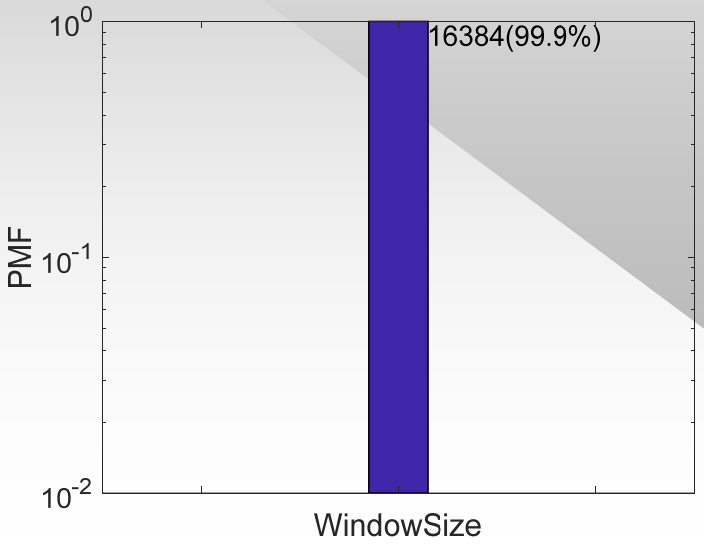


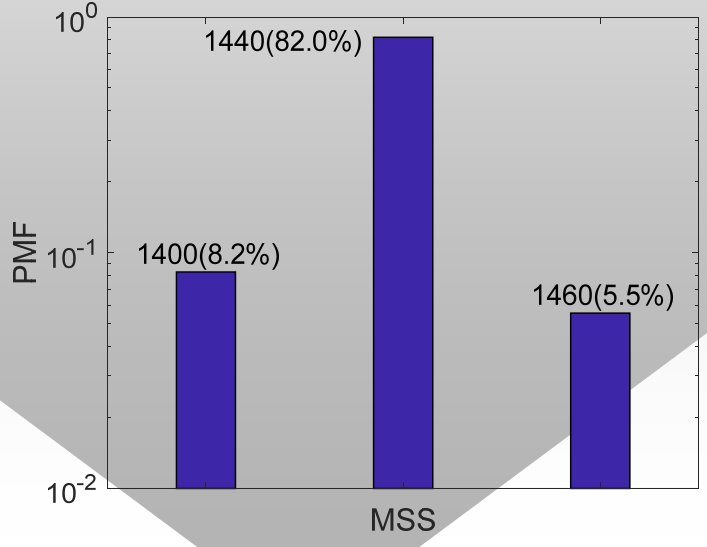


Internet Scan

- Faulds outputs interesting details about end-user behavior in modifying each device in the database

Dell Printers





Internet Scan

- Faulds found several industrial and enterprise devices reachable from the Internet

OS	Count	Type
Polycom HDX 8000	266 K	Video Conf.
Hickman ITV 450D	67 K	Video Conf.
Cisco IP Phone 7900	27 K	IP Phone
AVTech RoomAlert	21 K	Cyberphysical
Loytec Lighting Control	20 K	Cyberphysical
Tandberg Codian MCU	20 K	Video Conf.
Polycom Realpresence	18 K	Video Conf.
AdTran IP Phones	11 K	IP Phone
D-Link Internet Camera	9 K	Video Conf.

OS	Count	Released
Windows 2000 / XP / 2003	1.51 M	2000 – 2003
Windows Server 2003 SP1	195 K	2005
Windows Server 2000 SP4	146 K	2003
FreeBSD 6.4	78 K	2008
Solaris 9 / 10	71 K	2003 / 2005
Mac OSX 10.4	36 K	2005
Novell Netware OES 2	1 K	2005

- We also see large numbers of old OSes that are no longer supported still online

Conclusion

- We introduced an iterative EM-based classifier called Faulds to improve single-probe fingerprinting
  - Outperforms previous best classifier Hershel+
  - Builds much more detailed output without increasing measurement cost
- Using an Internet scan, we use Faulds to produce world-wide OS measurements with exhaustive results

THANK YOU!