Challenges in OS Security

Daniela Oliveira
University of Florida

GREPSEC II Workshop, May 16-17, 2015

Challenges in OS Secu

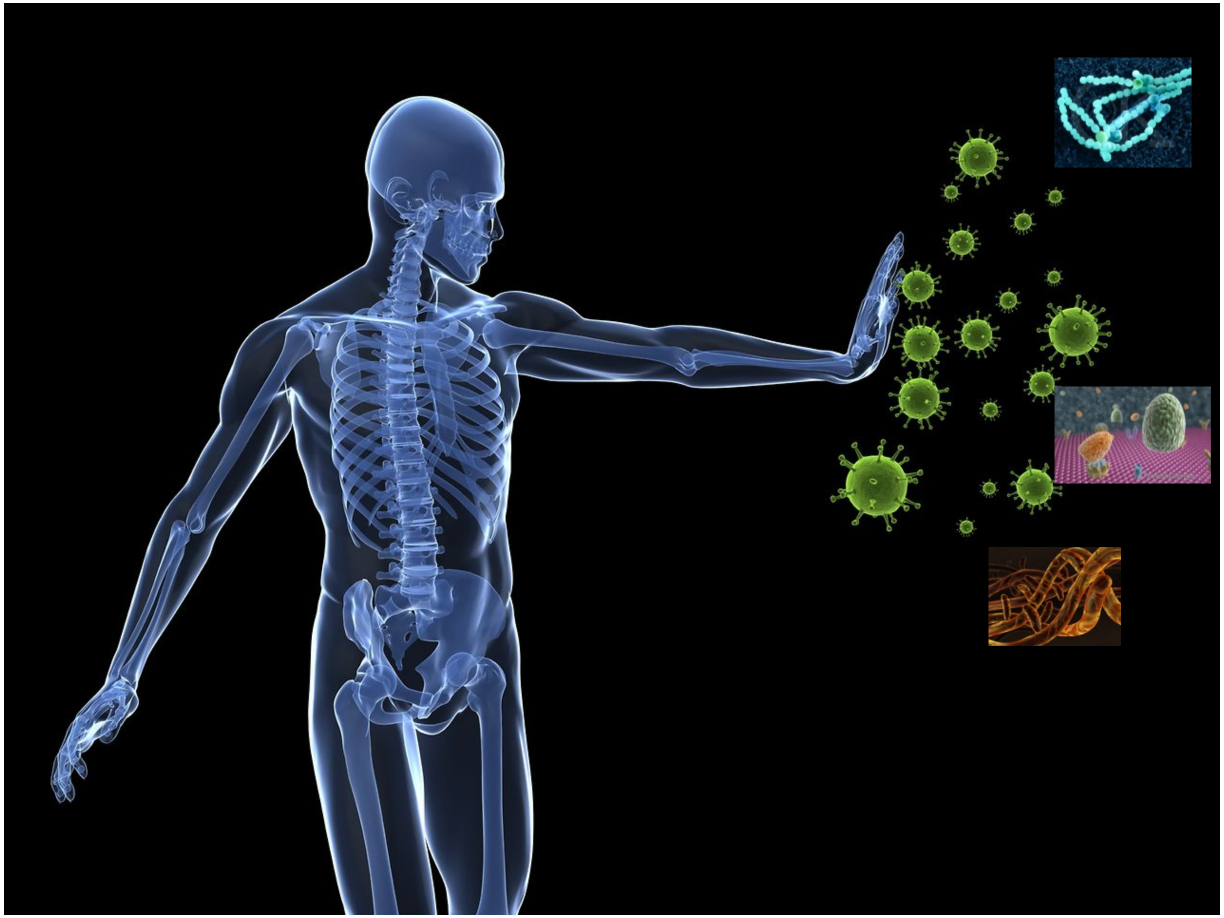
1. Safe co-existence with extensions
2. Collaboration with hardware
3. Overcoming monoculture

Challenges in OS Secu

1. Safe co-existence with extensions
2. Collaboration with hardware
3. Overcoming monoculture

Immune System

Cyber Security x The Mammalian Immune System



bacteria

viruses

fungi

parasites

toxins

Mammalian Immune System

Most successful defense system ever deployed

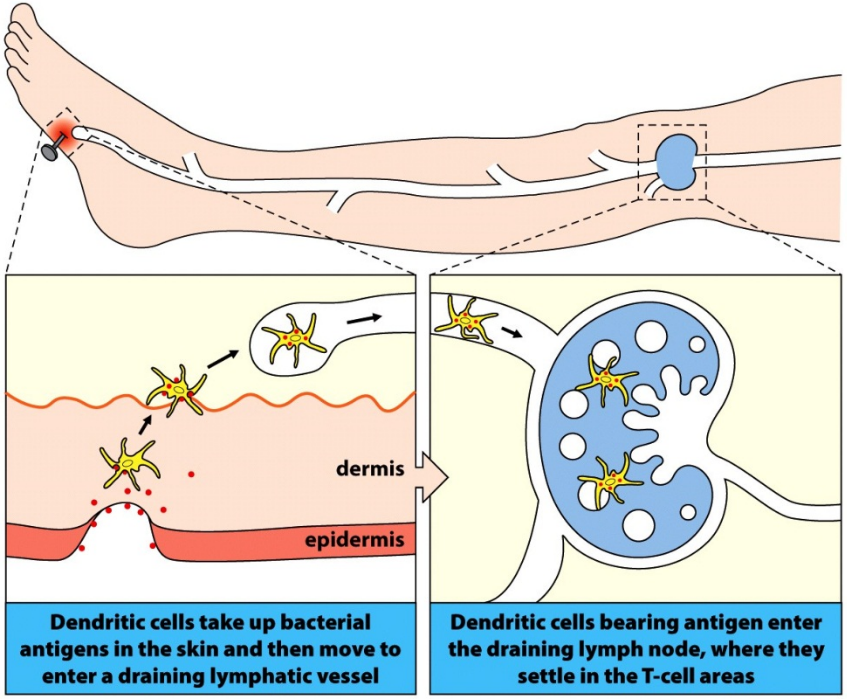
- Though it fails sometimes (cancer, auto-immune diseases, allergies)

Perfected by Nature over millions of years of evolution!

Mammalian Immune System

Employs high level of cooperation and communication among players

Maintains a symbiotic relationship with our microbiota



Dendritic cells take up bacterial antigens in the skin and then move to enter a draining lymphatic vessel

Dendritic cells bearing antigen enter the draining lymph node, where they settle in the T-cell areas

Figure 8.1 The Immune System, 3ed. (© Garland Science 2009)

Properties Lacking in Computer Security Approaches

Maintains a symbiotic relationship with our microbiota

Challenge 1

Employs high level of cooperation and communication among players

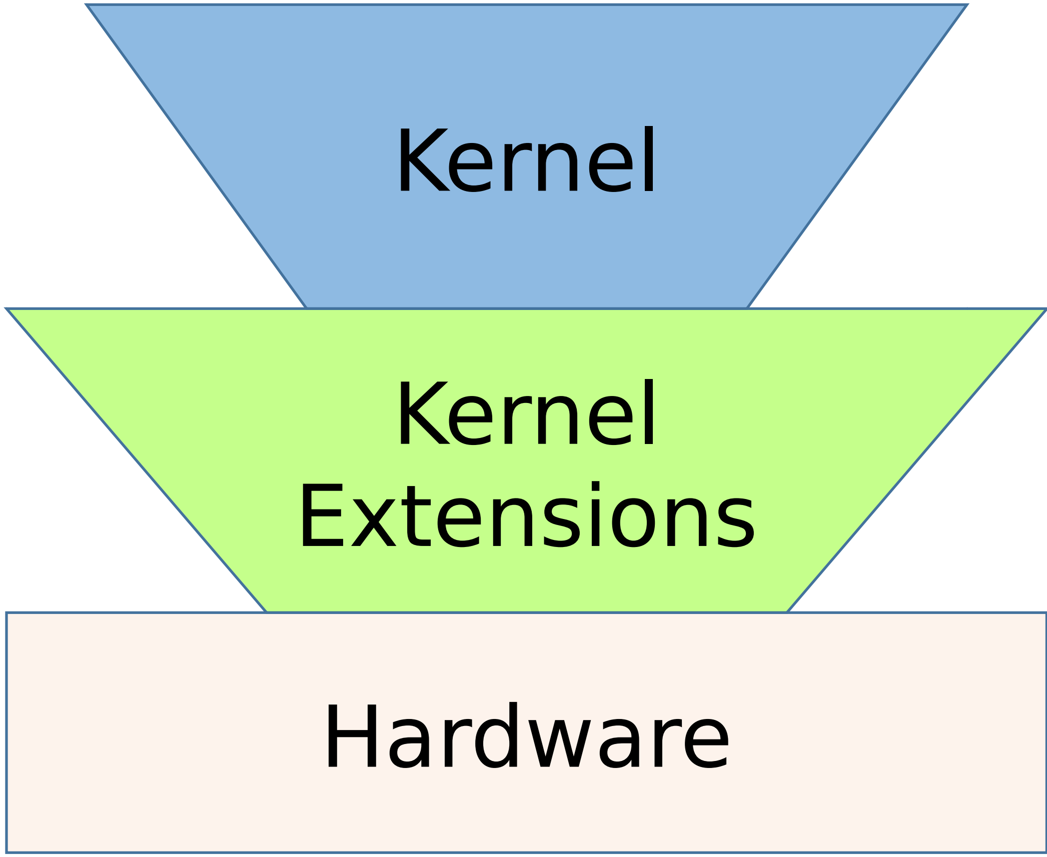
Challenge 2

Why don’t we leverage the immune system mechanisms in security approaches?

Safe Co-existence with Extensions

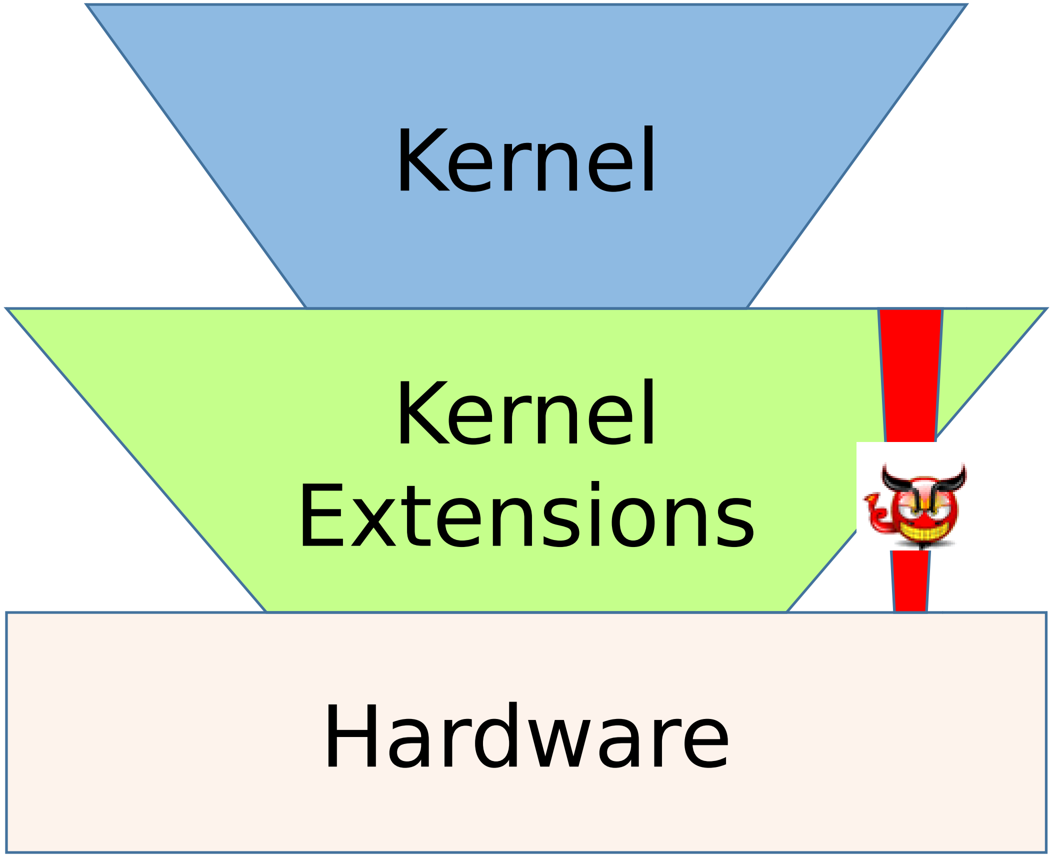
Kernel extensions represent at least 70% of kernel

Most benign and needed:



Kernel Extensions: Trusting the Untrusworthy

Small fraction is malicious



Untrustworthy Dependence - A Paradox?

OS must co-live with untrustworthy but *needed* extensions!

Untrustworthy Dependence

Immune system faces the same challenge:

Body made of more bacteria than human cells

Most benign and helpful:

- Digestion, obesity control, eczema, auto-immune diseases and allergy prevention

Small fraction cause pathologies

Untrustworthy Dependence

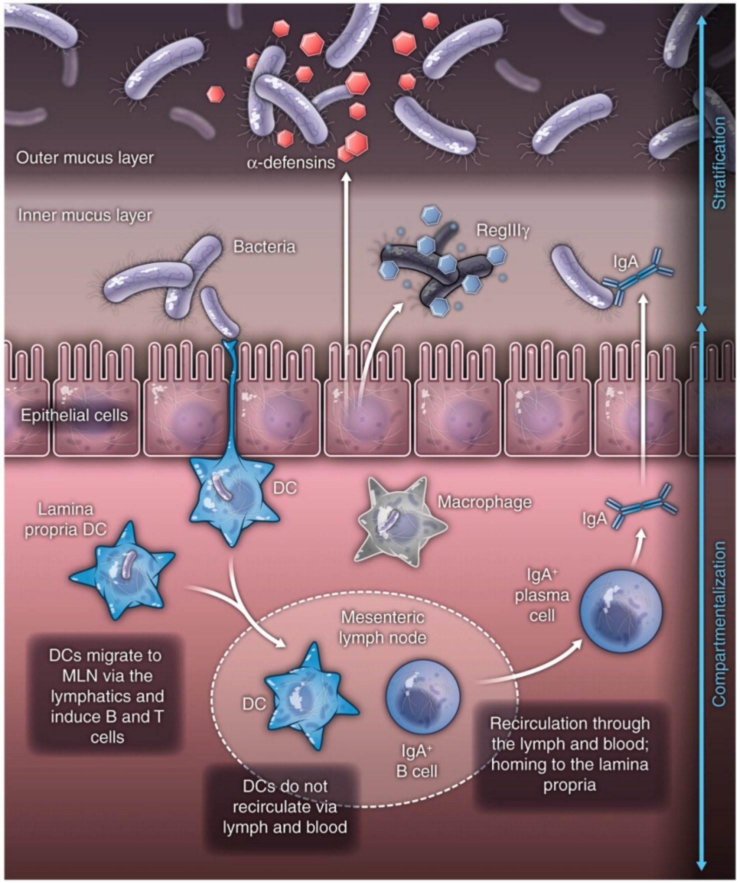
Immune evolved to maintain homeostatic relationship with microbiota:

Controlling microbial interactions with tissues

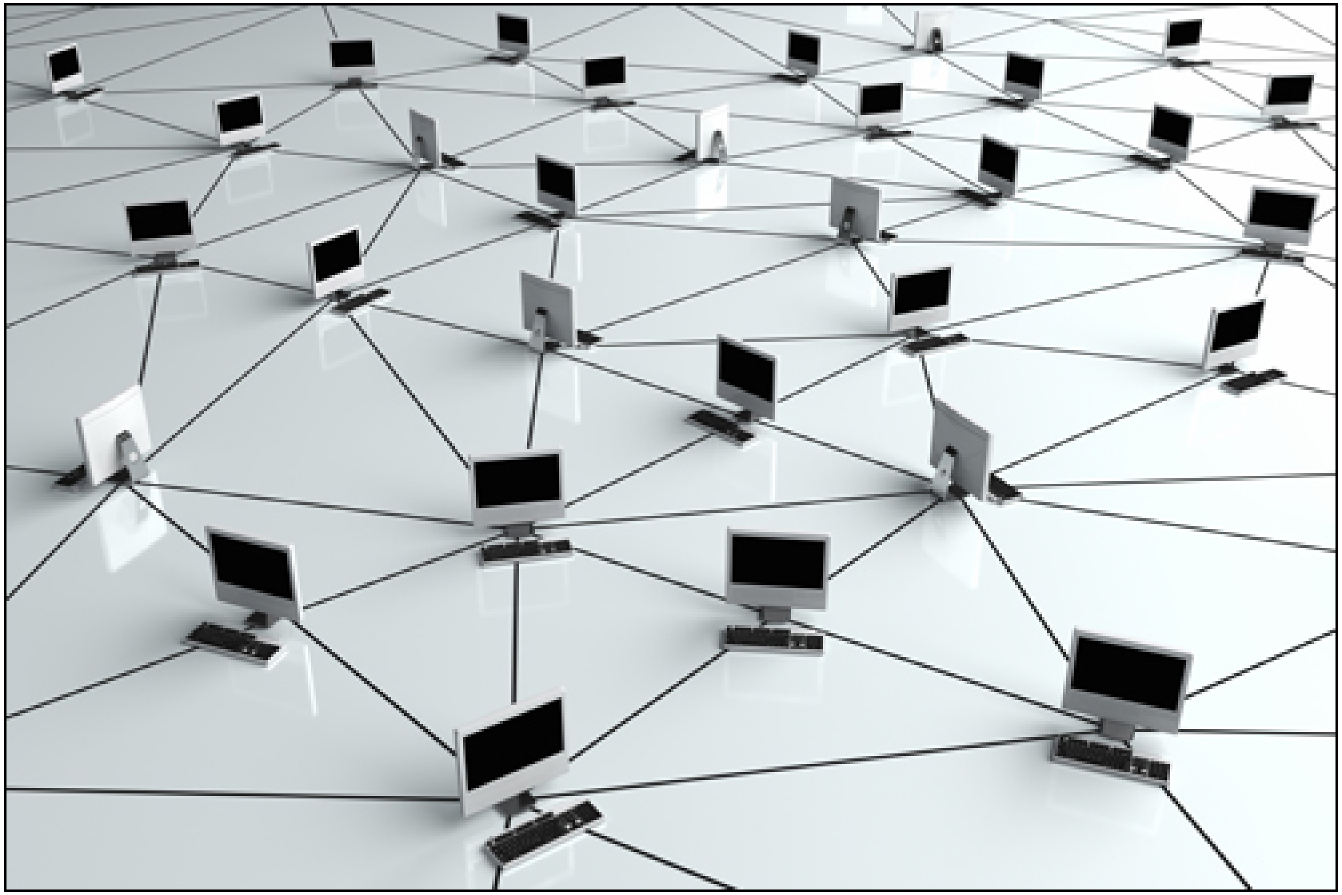
- Lessen potential for pathological outcomes

Immune System Approach

1. Confinement of bacteria to certain sites
2. Minimization of direct contact between bacteria and cell surfaces
3. Killing violating bacteria

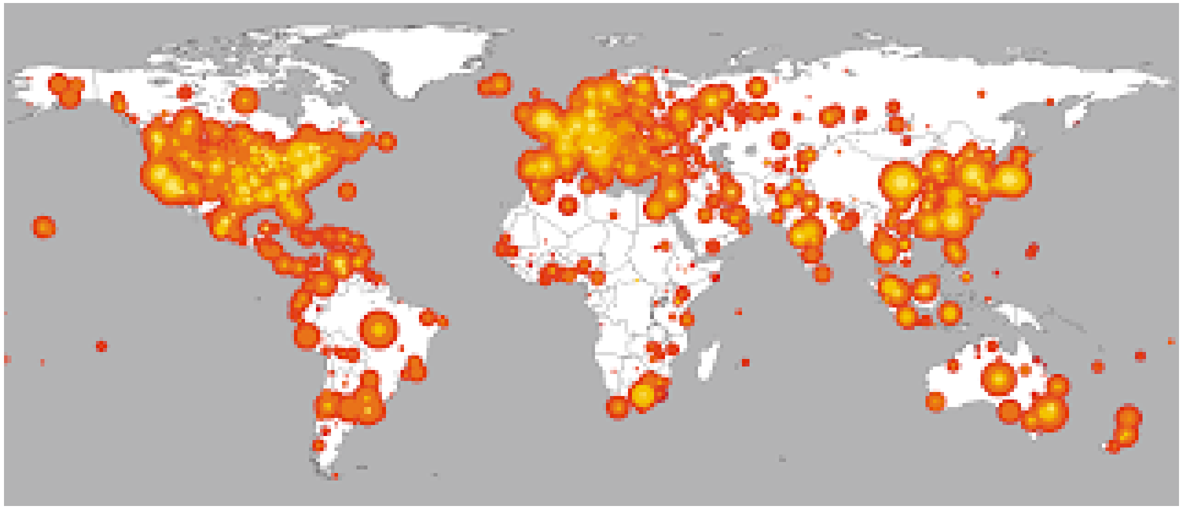


Challenge 3: Overcoming the Problems of Computer System Monoculture?



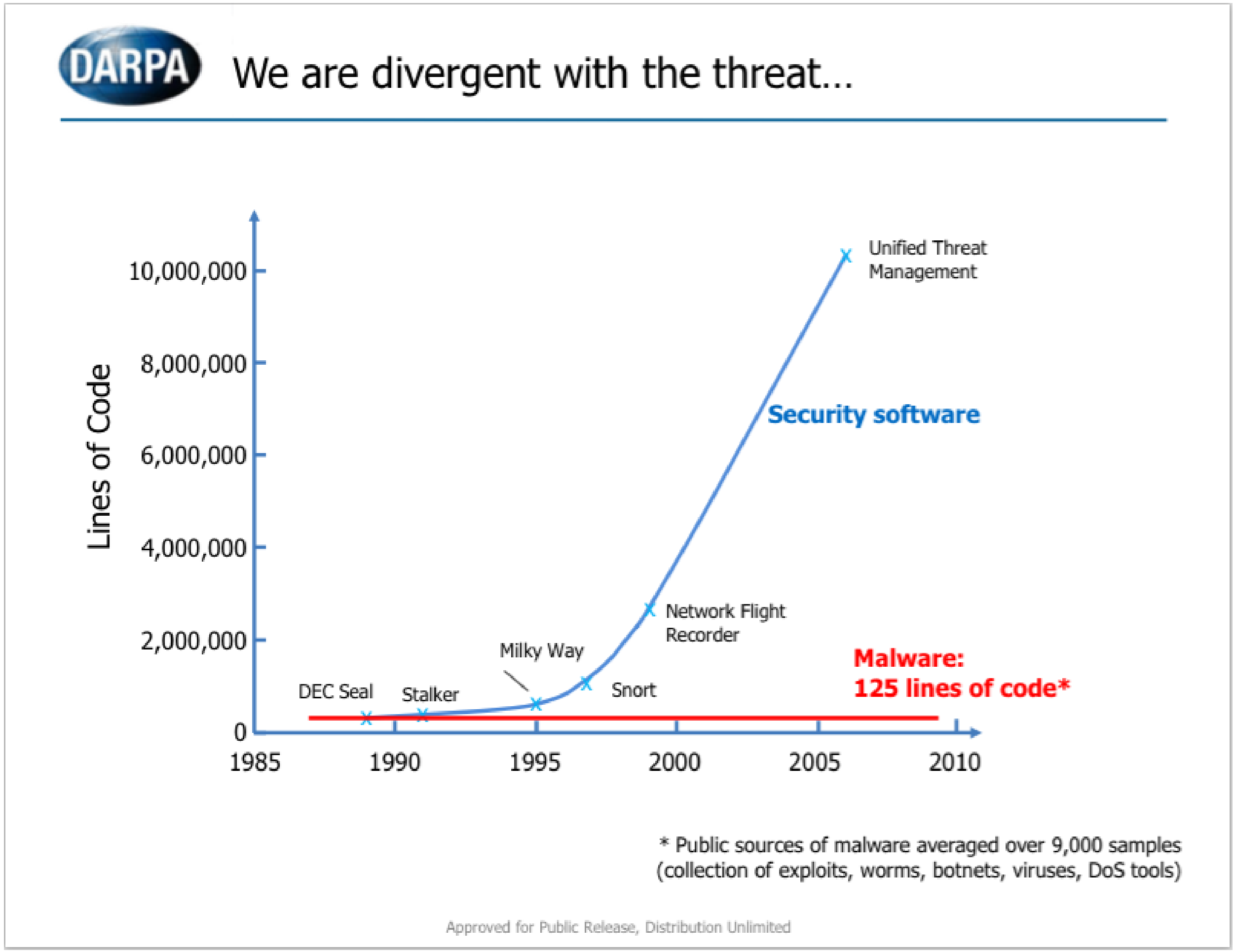
Predictability poses security problems…

Vulnerabilities exploitable on all systems of same type



- **Code Red 2001: 359,000 hosts infected**
- **$2 billion in losses**

Predictability Makes Attacker’s Life Easier



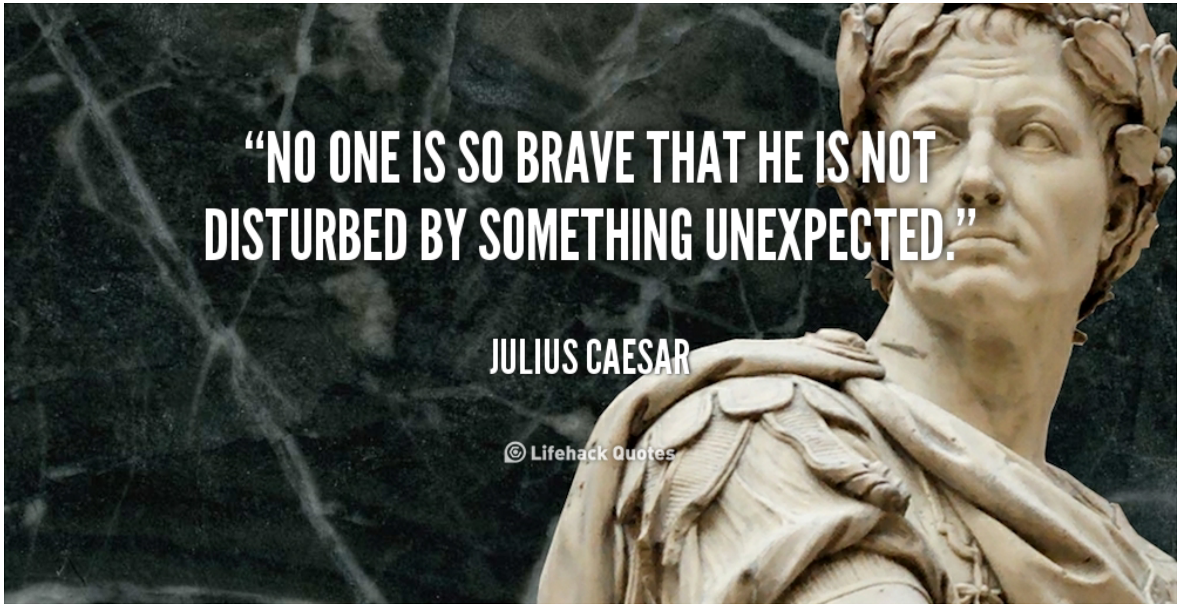
We are divergent with the threat…

* Public sources of malware averaged over 9,000 samples (collection of exploits, worms, botnets, viruses, DoS tools)

Approved for Public Release, Distribution Unlimited

Peiter “Mudge”: DARPA Framework for Cyber Security 2011

at If Operating Systems Were ***Trustwort*** ***Unpredictable***?



“NO ONE IS SO BRAVE THAT HE IS NOT DISTURBED BY SOMETHING UNEXPECTED.”

JULIUS CAESAR

© Lifehack Quotes

Unpredictability in Warfare – Battle of Salamis (480 B.C)

Unpredictability “Trends”

Address Space Layout Randomization (ASLR)

ISA Randomization

Compiler Specialization

Diverse implementation

N-version programming, library OSes

Still residual certainty th
benefits attackers!

ariation without unpredictability is not enoug

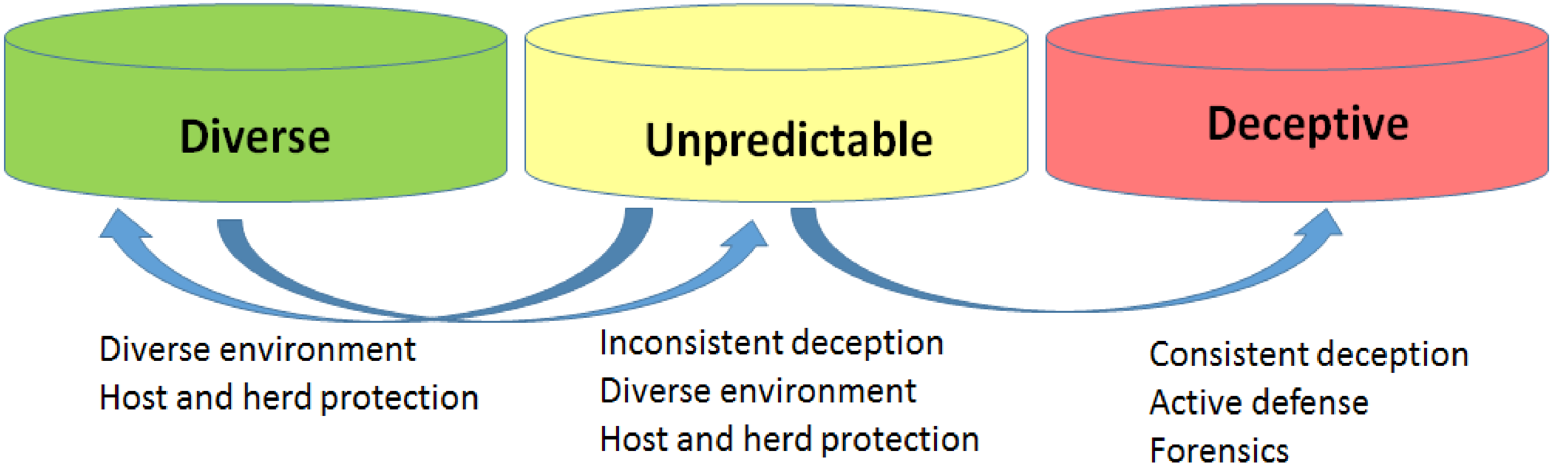
Trustworthy Unpredictability at OS Level

- For “good” uses: OS is predictable -> efficiency and reliability
- For “bad” uses: OS inefficient and unreliable

Selective Unpredictability

Spectrum Behavior O

Chameleon



Diverse

Unpredictable

Deceptive

Diverse environment
Host and herd protection

Inconsistent deception
Diverse environment
Host and herd protection

Consistent deception
Active defense
Forensics

Typical Scenario

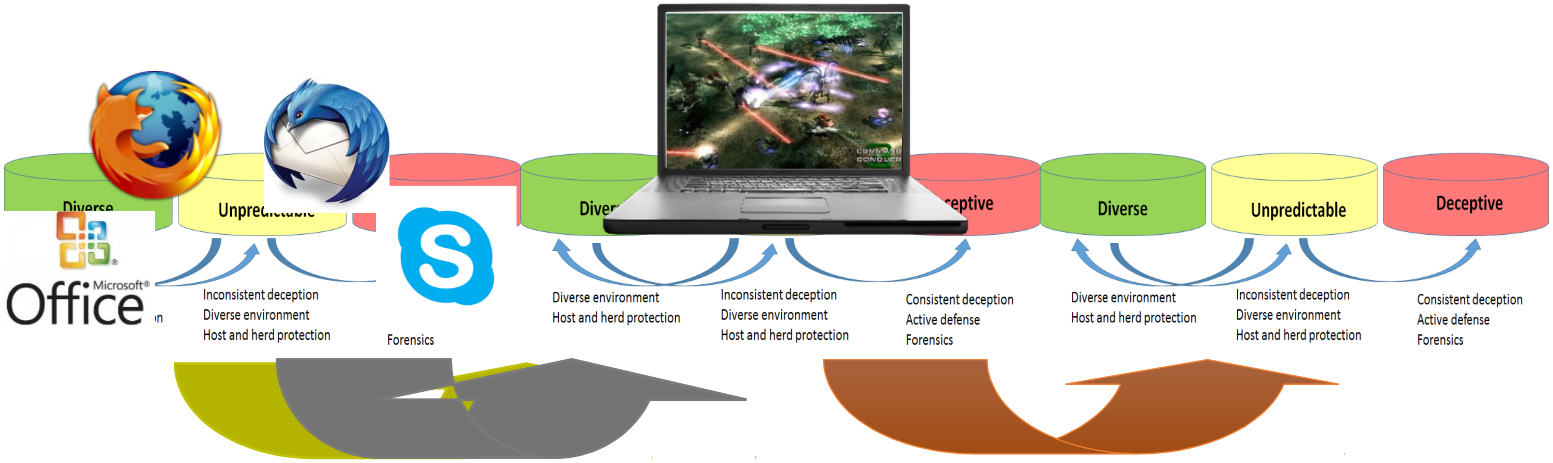
Bob, 78, living in a retirement community in Florida

not computer savvy, clicks in links from phishing email, installing malware

Malware engage in later DDoS attacks

Bob never notices: malware is active only after 1am.

Chameleon Scenario



Inconsistent deception
Diverse environment
Host and herd protection

Forensics

Diverse environment
Host and herd protection

Inconsistent deception
Diverse environment
Host and herd protection

Consistent deception
Active defense
Forensics

Diverse environment
Host and herd protection

Inconsistent deception
Diverse environment
Host and herd protection

Consistent deception
Active defense
Forensics

Preliminary Work

Assumptions:

- Malware is usually poorly written
- Robust applications have end-to-end checks

Methodology

- Use of ptrace to introduce unpredictability at system call level

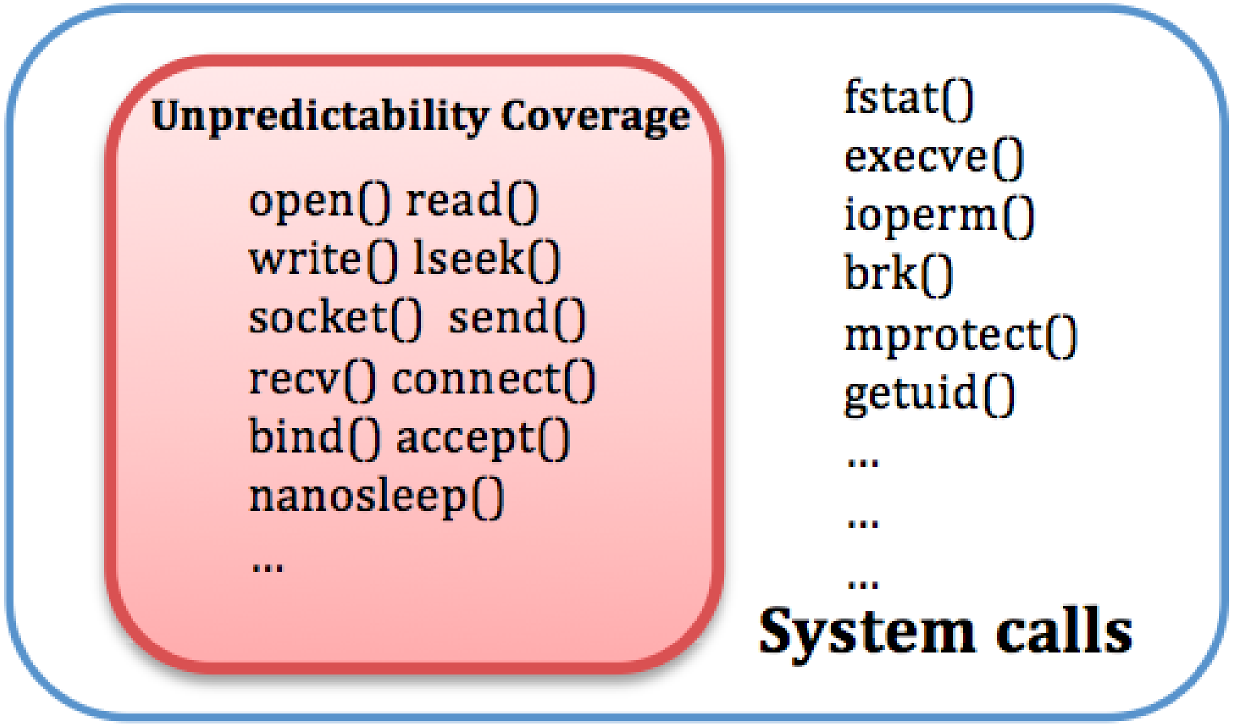
R. Sun, D. Porter, D. Oliveira and M. Bishop. The Case for Less Predictable Operating System Behavior. 15th Workshop on Hot Topics in Operating Systems (HotOS). Kartause Ittingen, Switzerland, May 18-20 2015

Strategies

Strategy 1: Silence the system call
Strategy 2: Change buffer bytes
Strategy 3: Add more wait time
Strategy 4: Change file pointer

Unpredictability Coverage

Only for system calls not critical to process start-up



Unpredictability Coverage

open() read()
write() lseek()
socket() send()
recv() connect()
bind() accept()
nanosleep()
...

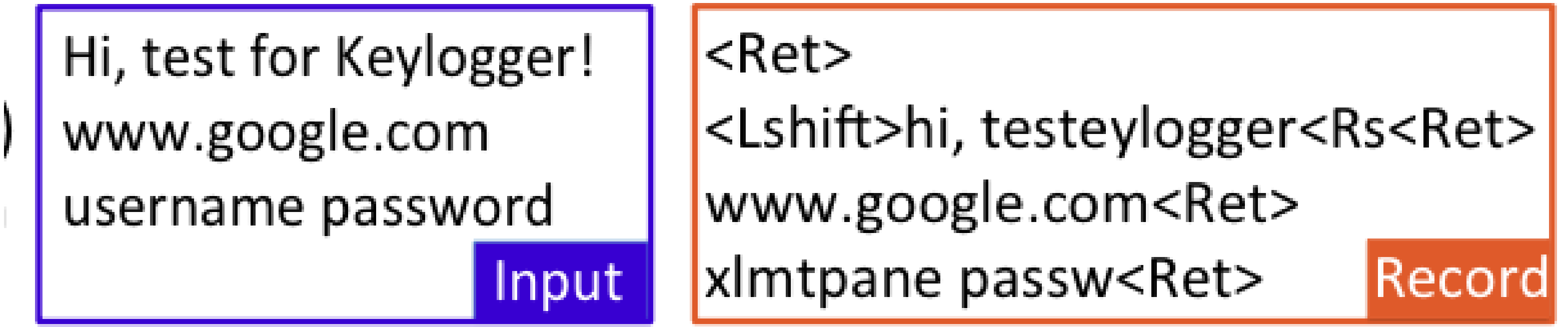
fstat()
execve()
ioperm()
brk()
mprotect()
getuid()
...
...
...

System calls

Keylogger with Unpredictability

Strategies:

- Change `write( fd, *buf, size)` buffer;
- Change `lseek( fd, offset, whence)` pointer;



Hi, test for Keylogger!
www.google.com
username password

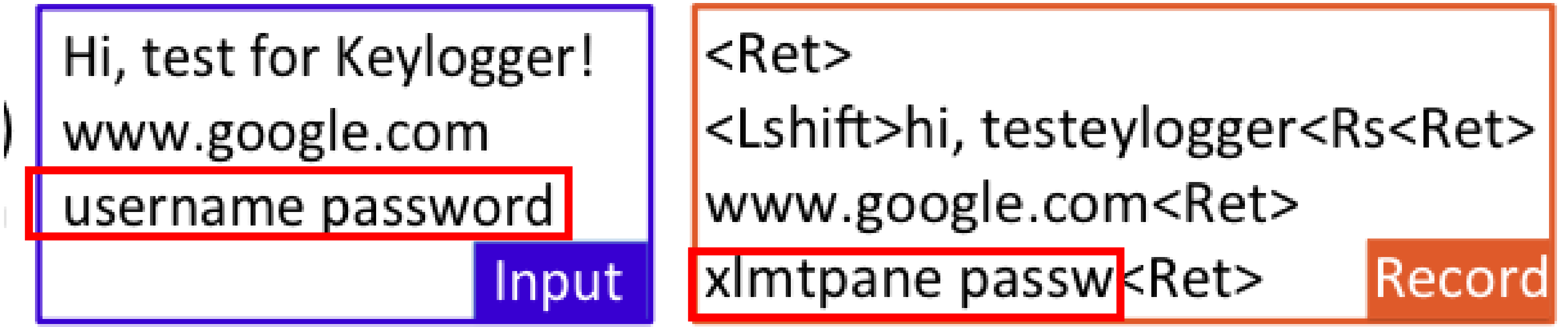
<Ret>
<Lshift>hi, testeylogger<Rs<Ret>
www.google.com<Ret>
xlmtpane passw<Ret>

Keylogger with Unpredictability

Strategies:

Change `write( fd, *buf, size)` buffer;

Change `lseek( fd, offset, whence)` pointer;



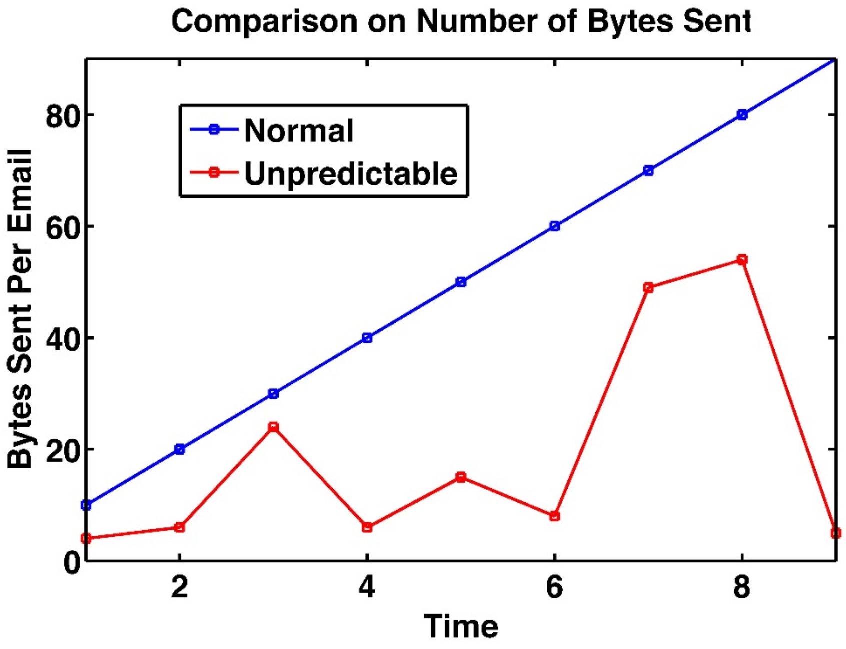
Hi, test for Keylogger!
www.google.com
username password

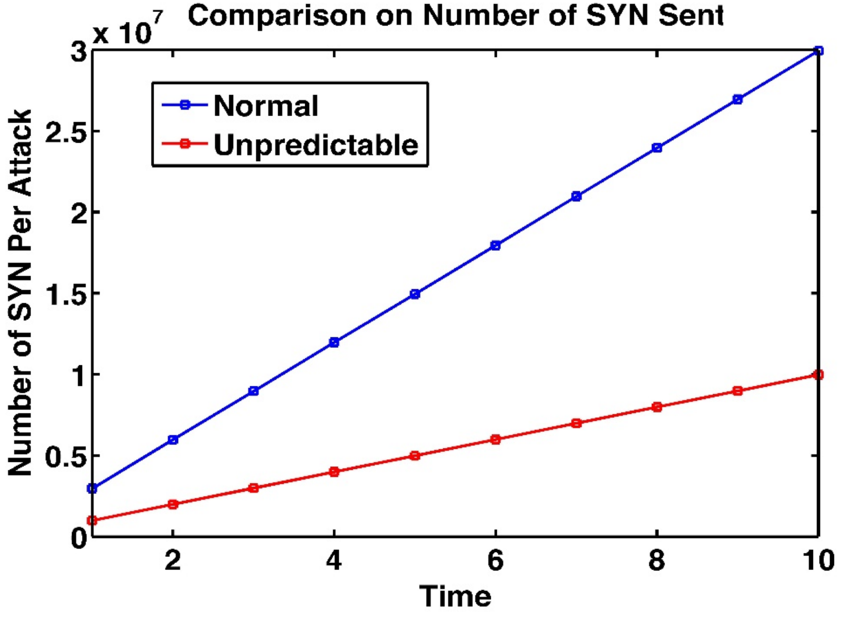
<Ret>
<Lshift>hi, testeylogger<Rs<Ret>
www.google.com<Ret>
xlmtpane passw<Ret>

Botnet with Unpredictability

Strategies:

- Silence `read( fd, *buf, size);`
- Silence or reduce len in `sendto( sockfd, *buf, len, …);`





What About Benign Software?

Firefox, Thunderbird and Skype

Works normally most of time
Occasional warnings
Functionalities temporarily unavailable

Concluding Remarks

Holy grail of system design: thwart attacker with less effort than generating attacks

Chameleon makes systems diverse by design and actively secure:

- Diverse + Unpredictable: every instance of system behaves differently
- Deceptive: lures adversaries into revealing their strategies

Unpredictability is promising!

Collaboratc

Ruimin Sun, PhD Student University of Florida

Don Porter
Stony Brook

Matt Bishop
UC Davis

Natalie Ebner
University of Florida

University of Florida is Rising!

Patrick Traynor
Mobile Security

Juan Gilbert
Electronic Voting

Mark Tehranipoor
Supply Chain Security

Kevin Butler
Cyber Physical Systems

Tom Shrimpton
Crypto

Domenic Forte
Hardware Trojan Prevention

Swarup Bhunia
Hardware Trojan Detection

Damon Woodard
Biometrics/fingerprinting

Thank you!

daniela@ece.ufl.edu