Nexus: An Operating System for Trustworthy Computing

Alan Shieh  Dan Williams  Kevin Walsh
Emin Gün Sirer  Fred B. Schneider

Department of Computer Science
Cornell University
New hardware for trustworthy computing is emerging

How best to exploit this new hardware?
Project Overview

- **Nexus OS**
  - Builds on Trusted Platform Module (TPM)
    - Industry-standard secure coprocessor
    - Simple and pervasive
  - Provides new trustworthy computing abstractions
  - Provides assurance through a small TCB
  - Enables new trustworthy applications
TPM primitives

- Hardware root of trust
- Functionality:
  - Data integrity
  - Key storage
  - Attestation (expects hashes)

- Reasonable starting point...
TPM limitations

- Mismatch between TPM and application needs
  - Holds only a few secrets & keys
  - Attests to a system snapshot
  - Supports only hash-based authentication and authorization
Extant OS limitations

- Existing OSes are not suited for trustworthy computing
  - Linux and Windows simply too big
  - Monolithic architecture → violates principle of least privilege
  - No strong isolation between components
Nexus: A New OS

- Nexus OS bridges the gap
  - Generalizes and virtualizes the TPM
  - Enables authorization from semantic properties
Nexus: A New OS

- Supports new abstractions with comparable level of assurance relative to TPM
  - Small TCB
    - Exclude drivers and services (user-level)
    - Exclude secondary storage
  - Fine-grain components → restricted policies
  - Strong isolation of components

- ... Respectable performance too!
New abstractions

- **Secure memory regions** with mandatory access control and persistence
- **Active attestation** attests to a component’s properties and environment.
  - Assigns a descriptive label to component
New Nexus OS abstraction:

Active attestation labels

- **Labeling functions** generate meaningful, flexible labels from:
  - Result of analysis / PCC
  - Use of reference monitors
  - Run in execution environment

- Unlike hash, captures only property of interest

- Used pervasively in the Nexus
  - E.g., IPC binding & invocation, access to secure memory regions, etc.
New Nexus OS abstraction:

Secure memory regions

- Secure memory regions are used to store sensitive application data
- Guarantees:
  - Integrity
  - Confidentiality
  - Persistence
Secure memory regions

- Admit application-specific optimizations
  - Use knowledge of access patterns to compute optimal block size for hash-trees
    [Williams & Sirer, TNC2004]

- Invaluable for user-level services
  - E.g. Linear capability manager
  - ... or any history-dependent policies (via security automata)
Status of the Nexus OS

Working prototype of kernel and new abstractions

Isolated Protection Domains
- MPlayer
- E-mail
- Keymgr
- TCP

Unprivileged
- Apache
- Pine
- exim
- ssh
- Linux compat

Privileged
- Nexus user interface
- Nexus driver interface
- IPC
- Trusted computing

Safe device access
- Video driver
- Network driver
- Audio driver
Applications

- Working applications
  - DRM-compliant media player
  - Spam-proof e-mail system
  - Tamper-evident system log
  - Attested MACEDON application

- Real applications provide insights that drive investigation into active attestation
Media player example

- Secure memory regions protect movie data and policy metadata
- Linear capabilities restrict media to a limited number of plays
- Active attestation attests to future behavior of media player
  - Media player does not write to disk
    - This property describes a family of media players
Media player example

Disk

Media storage

Secure memory region encrypts movie

Linear capability manager

Secure memory region protects against tampering of capabilities
Media player example

- Media storage
  Secure memory region encrypts movie

- Linear capability manager
  Secure memory region protects against tampering of capabilities

- Disk

- Movie company

- Download data and install policy
Media player example

Media storage
Movie.avi
Secure memory region encrypts movie

Linear capability manager
Can'tWriteDisk => Movie.avi
RightsHolder => Modify()
Secure memory region protects against tampering of capabilities

Disk

Save data

Download data and install policy

Movie company
Media player example

Media storage
Movie.avi
Secure memory region encrypts movie

Linear capability manager
Can'tWriteDisk => Movie.avi 5
RightsHolder => Modify()
Secure memory region protects against tampering of capabilities

Movie company

NoDiskIO
reference monitor

Disk
Media player example

NoDiskIO
reference monitor
Media player

Label: Can'tWriteDisk

Media storage
Movie.avi
Secure memory region encrypts movie

Linear capability manager
Can'tWriteDisk => Movie.avi 5
RightsHolder => Modify()
Secure memory region protects against tampering of capabilities

Movie company
Media player example

NoDiskIO reference monitor

Media player

Label: Can'tWriteDisk
Request movie

Media storage

Movie.avi

Secure memory region encrypts movie

Linear capability manager

Can'tWriteDisk => Movie.avi 5

RightsHolder => Modify()

Secure memory region protects against tampering of capabilities

Movie company
Media player example

**WARNING**
Federal law provides severe civil and criminal penalties for the unauthorized reproduction, distribution or exhibition of copyrighted motion pictures, video tapes, or video disc.

Criminal copyright infringement is investigated by the FBI and may constitute a felony with a maximum penalty of up to five years in prison and/or a $250,000 fine.

---

**Media storage**
- Movie.avi
- Secure memory region encrypts movie

**Linear capability manager**
- Can'tWriteDisk => Movie.avi
- RightsHolder => Modify()
- Secure memory region protects against tampering of capabilities

**NoDiskI0 reference monitor**
- Media player
- Label: Can'tWriteDisk

---

**Send data to player**
- Media player
- Play movie

**Supply movie data**
- Media storage

---

**Movie company**
Media player example

- **WARNING**: Federal law provides severe civil and criminal penalties for the unauthorized reproduction, distribution or exhibition of copyrighted motion pictures, video tapes, or video discs. Criminal copyright infringement is investigated by the FBI and may constitute a felony with a maximum penalty of up to five years in prison and/or a $250,000 fine.

- **NoDiskIO reference monitor**
  - Media player
  - **Label**: Can'tWriteDisk
  - Request movie
  - **Send data to player**
    - Linear capability manager
      - Can'tWriteDisk => Movie.avi x 4
      - RightsHolder => Modify()
      - Secure memory region protects against tampering of capabilities

- **Media storage**: Movie.avi
  - Secure memory region encrypts movie

- **Movie company**

- **Play movie**
  - Supply movie data
  - Disk
Media player example

**WARNING**
Federal law provides severe civil and criminal penalties for the unauthorized reproduction, distribution or exhibition of copyrighted motion pictures, videotapes, or videodiscs.

Criminal copyright infringement is investigated by the FBI and may constitute a felony with a maximum penalty of up to five years in prison and/or a $250,000 fine.

---

**NoDiskI0 reference monitor**

**Disk**

**Media player**

**Media storage**

*Movie.avi*

Secure memory region encrypts movie

---

**Linear capability manager**

Can'tWriteDisk => Movie.avi X 4

RightsHolder => Modify()

Secure memory region protects against tampering of capabilities

---

**Movie company**

---

*Label: CanWriteDisk*
Media player example

Media storage

Movie.avi

Secure memory region encrypts movie

NoDiskIO reference monitor

Media player

Disk

Label: CanWriteDisk

Request movie

Linear capability manager

Can'tWriteDisk => Movie.avi  X 4

RightsHolder => Modify()

Secure memory region protects against tampering of capabilities

Movie company
Media player example

Media storage

- Movie.avi

Secure memory region encrypts movie

NoDiskIOWarning

reference monitor

Media player

Disk

Linear capability manager

Can'tWriteDisk => Movie.avi × 4

RightsHolder => Modify( )

Secure memory region protects against tampering of capabilities

Set new count

Movie company

Hacker
Media player example

**WARNING**

Federal law provides severe civil and criminal penalties for the unauthorized reproduction, distribution or exhibition of copyrighted motion pictures, video tapes, or video discs.

Criminal copyright infringement is investigated by the FBI and may constitute a felony with a maximum penalty of up to five years in prison and/or a $250,000 fine.

---

**Media storage**

*Movie.avi*

Secure memory region encrypts movie

---

**NoDiskIO reference monitor**

**Disk**

---

**Linear capability manager**

*Can'tWriteDisk* => *Movie.avi X 4*

*RightsHolder* => *Modify()

Secure memory region protects against tampering of capabilities

---

**Label: RightsHolder**

*Movie company*

---

**Label: Unknown**

*Hacker*

---

Set new count
Media player example

Warning: Federal law provides severe civil and criminal penalties for the unauthorized reproduction, distribution or exhibition of copyrighted motion pictures, video tapes, or video discs. Criminal copyright infringement is investigated by the FBI and may constitute a felony with a maximum penalty of up to five years in prison and/or a $250,000 fine.

Media player

NoDiskIO reference monitor

Disk

Media player

Media storage

Movie.avi

Secure memory region encrypts movie

Linear capability manager

Can'tWriteDisk => Movie.avi X 4

RightsHolder => Modify()

Secure memory region protects against tampering of capabilities

Label: RightsHolder

Movie company

Label: Unknown

Hacker

Set new count
Spam-proof e-mail

- Only “non-spam” e-mail clients can sign message with special key

- “Non-spam” clients:
  - Client binary is approved
  - User has typed in text during this execution
Nexus lessons

- Active attestation captures application properties
- Attesting to properties enables meaningful authorization
- Third-parties can provide tools for extracting and enforcing properties
Trustworthy computing requires new properties from OS
The Nexus is a new OS for trusted computing
- Capture the semantic properties of programs
- Provide assurance about future behavior
There are many opportunities for future research
- New tools for capturing properties
- New applications that require additional trust