RESEARCH

Retrospective: 30 Years of Cybersecurity R&D

Stephen Smalley
NSA Laboratory for Advanced Cybersecurity Research (LACR)

April 3, 2024

Did you know that…

- All modern Android devices use a security framework first developed by NSA’s Laboratory for Advanced Cybersecurity Research (LACR).
  - And so do many Linux-based systems.
- All iOS devices run a security framework whose development was originally sponsored by LACR.
  - And so do macOS and FreeBSD-based systems.
- Windows Virtualization-based Security embodies multiple concepts from a Secure Virtual Platform architecture first created by LACR.
- How did we get there?

About the Laboratory for Advanced Cybersecurity Research (LACR)

- Originally created as a dedicated research organization in 1990.
  - Although NSA was doing computer security research decades before.
- R&D in support of NSA’s Cybersecurity mission to protect National Security Information and Systems.
- First at NSA to create and release open source software – SELinux, 2000.
- Long history of open source contribution and collaboration.
  - Linux, Xen, FreeBSD, Darwin, OpenSolaris, Android, Zephyr
- With both direct and indirect impacts on real systems, both open source and proprietary.

Thirty Years Ago…

- I was a relatively new hire into the OS security research team in LACR.
- Linux 1.0 was just released (Linus: “A better UNIX than Windows NT”).
- Google didn’t exist (and wasn’t a verb!).
- No mainstream operating system supported Mandatory Access Control.
- The Trusted Platform Module (TPM) hadn’t even been specified yet.
- Cloud computing (as we know it today) didn’t exist.
- Hardware virtualization wasn’t yet supported by commodity processors.
- Smartphones didn’t exist (unless you count the IBM Simon!).
- Trusted Execution Environments were not even a concept.
- AI/ML was…slightly less advanced.

In the beginning…

- The 1990s: “Peace, Prosperity and the Internet” (history.com)
- *Synergy: A Distributed, Trusted, Microkernel-based Operating System,*1993
  - Distributed Trusted Mach (DTMach)
  - The Distributed Trusted Operating System (DTOS)
  - The Flux Advanced Security Kernel (Flask)
- The start of recurring themes for our research
  - Microkernels for security and security for microkernels
  - Flexible security/Mandatory Access Control (MAC): no one size fits all
- *The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments*, 21st NISSC, Oct 1998.

NSA + Open Source = SELinux?

- The trials and tribulations of research prototypes & technology transfer
- National Security Council recommendation
- Our Goals
  - Demonstrate viability of security architecture in a real OS
  - Provide an open reference implementation
  - Provide a long-term research platform (still going strong after 23 years!)
- Linux as an emerging platform
- Developing the code was the easy part! Initial prototype created in 1999.
- First public release: December 22, 2000, based on Linux 2.2.

Growing Up

- A community quickly coalesces around SELinux.
- Multiple rewrites to make it acceptable: Third time's a charm!
- SELinux upstream merge in Linux 2.6.0-test3, Aug 2003.
- Linux 2.6.0 ("The beaver is out of detox") released Dec 2003.
  - 20 years of SELinux in the mainline Linux kernel!
- Integration into a GNU/Linux distribution
  - 2004: Fedora Core 3; 20 years of SELinux in Fedora!
- Extending upward into middleware and applications.

Branching Out

- In parallel with our work to mature and extend SELinux.
- Co-sponsored flexible MAC development for FreeBSD and Darwin.
    - Adopted into FreeBSD (experimental in 2003, default in 2009).
    - Leveraged earlier DTMach/DTOS microkernel R&D for Darwin.
    - Adopted into macOS (2007) and iOS (2008) for app sandboxing.
- Joint development of OpenSolaris Flexible MAC (FMAC), 2007-2009.
    - RIP OpenSolaris 2010

Going Virtual

- NetTop, starting circa 2000
    - VMWare/SELinux hybrid to support multiple security level connectivity from a single desktop
    - *NetTop Eight Years Later*, The Next Wave, 2008
- Secure Virtual Platform (SVP), starting early 2000s
    - Explored emerging hardware virtualization and trusted computing paradigms to address residual risks
    - Applied these technologies to construct a secure system architecture
    - *Secure Virtual Platform Research*, OpenXT Summit 2016

Hypervisors: Microkernels Revisited

- Opportunity to revisit microkernel-like OS architecture for security
  - Isolate untrusted and security-critical components
  - Enforce assured pipelines, e.g. inline VPN or DAR
- Xen chosen as a research platform
  - “type 1” hypervisor, community, adoption, open source
- Securing virtualization
  - Hypervisor MAC – XSM/Flask first merged 2007, full support in 2013
  - Dom0 disaggregation – *Breaking up is hard to do: Security and functionality in a commodity hypervisor*, SOSP’2011
  - Secure IVC – OpenXT v4v (2011/14), Xen Argo (2019)

Trust but Verify

- Recognized Trusted Platform Module (TPM) as a key enabling technology
  - Verifiable, trustworthy report of loaded software and configuration
  - Protection of long term secrets from leakage and misuse
  - Resilient even in the face of complete software compromise
- But also recognized the remaining gaps and challenges
  - Scalability, flexibility, dynamism, chain of trust
  - Virtualization support
  - Need for runtime integrity measurement
  - Need for flexible, layered attestations

Runtime Integrity: A Missing Link

- Invented technique for measuring and appraising the integrity of running software: contextual inspection.
- Prototyped for:
  - Linux kernel (*Linux kernel integrity measurement using contextual inspection*, STC’2007)
  - Xen hypervisor (*STM/PE & XHIM*, PSEC’2018)
  - Windows kernel
- Just now becoming generally available in commercial products.
- Zero Trust for operating systems / hypervisors

Finding a Place to Stand

- Need for hardware roots of trust for load-time and run-time integrity measurement
  - Dynamic Root of Trust for Measurement – TXT/SVM
  - SMI Transfer Monitor (STM)
- *Using the Intel STM for Protected Execution*, PSEC 2018
- *Implementing STM Support for Coreboot*, OSFC 2019
- SMM isolation and SMI de-privileging finally entering the mainstream

Flexible Attestation

- System architectures to support comprehensive, flexible load-time and runtime measurement.
- Flexible support for selective, policy-driven attestations.
- Protocols for attestation.
- Demonstrated in Maat open source framework for orchestrating flexible, layered attestations.
  - First described in *Attestation: Evidence and Trust*, ICICS’08
  - *Flexible Mechanisms for Remote Attestation*, ACM Trans. Priv. Sec. 2021.
  - Open source release in 2022.

Going Mobile

- Enhancing mobile OS security: SE (for) Android
  - Open source release in 2012, adoption beginning in 2013
  - *Security Enhanced Android: Bringing Flexible MAC to Android*, 2013
  - A decade of SE for Android, running on > 3 billion active devices
- SVP for mobile devices: secure wireless laptop, smartphone virtualization
  - Influenced XenClient XT / OpenXT
  - Influenced Samsung's Knox architecture
  - *Laying a Secure Foundation for Mobile Devices*, NDSS'13

The s in IoT stands for Security

- Spanning the gamut from Linux-based operating systems to Zephyr to Fuchsia
  - Yocto, Android Things, Zephyr, Fuchsia
  - *Security in Zephyr and Fuchsia*, Linux Security Summit 2018
- Adapting to microcontroller hardware
  - MPUs vs MMUs
  - TrustZone-M vs TrustZone
  - CHERI for microcontrollers

Shrink the TCB: Use a TEE

- Early R&D into using Arm TrustZone for mobile devices
  - Place to host TPM/MTM-like functionality, runtime integrity
- Intel SGX fundamentally changed the threat model - 2013
  - Opportunity to shrink Trusted Computing Base (TCB) to a portion of the application
- Trend toward VM-based Trusted Execution Environments (TEEs)
  - AMD SEV-SNP, Intel TDX, Arm Realms
  - With corresponding expansion of the TCB

Securing the Cloud

- Growing adoption and use of SELinux in cloud-focused Linux distributions
  - Bottlerocket Linux and Amazon Linux 2023
  - Azure Linux and Azure Boost
- The rise of confidential computing
  - Leveraging TEEs in the cloud
  - Enabling trustworthy AI/ML

The Persistent Relevance of the OS

- None of these technological advances have obviated the need for secure operating systems!
  - *The persistent relevance of the local operating system to global applications*, 7th ACM SIGOPS European workshop,1996.
- And OS security is not a static field.
  - SELinux itself is constantly evolving to address emerging needs and technologies.
    - And perhaps might even be replaced someday (hint: eBPF).
  - SVP/VBS-like architectures are now being proposed for Linux.

Questions?

- Contact me: sdsmall@uwe.nsa.gov
- SELinux Project, https://github.com/SELinuxProject
- NSA LACR, https://nsa.gov/LACR