马乐乐

更新于 2022 年 1 月

邮箱: lma03@wm.edu 电话: +86 135 5208 4690

兴趣 & 技能

工作兴趣
- **系统内核，系统安全，内存安全，权限分离;**
- **对云计算、边缘计算、IoT 设备的系统支持**。

工作技能
- 面向系统安全的软硬件协同设计 (CPU，编程语言，编译器，操作系统)；
- 进程内存安全（进程隔离，地址空间内部隔离，堆栈防护）；
- 虚拟化技术（虚拟机内省，反汇编，计算机取证，Docker 容器)；
- 编程语言: 熟练：C, C++, Bash; 熟悉：Go, Python, Java; 业余爱好: Coq, Haskell, Prolog.

示例作品
- 虚拟机内省的轻量虚拟机 TinyVMI (https://github.com/tinyvmi/tinyvmi/).
- 更多项目实践见主导/合作项目和学术论文。

教育状况

08/2015 – 01/2022 **博士，计算机科学与技术**
威廉玛丽学院，[美] 弗吉尼亚州威廉斯堡市
- 导师: 李群 教授，IEEE 院士（边缘计算、操作系统、深度学习、信息安全等）
- 博士论文：物联网环境下利用隔离技术提高系统安全与性能的研究

09/2012 – 07/2015 **硕士，计算机科学与技术**
中国科学院软件研究所，北京
- 导师：杨秋松 研究员（形式化方法、安全操作系统、计算机体系结构等）
- 硕士论文：用于内存安全监测的轻量虚拟机监控代理设计与实现

09/2008 – 06/2012 **学士，计算机科学与技术**
山东大学，山东省济南市
- 导师：孙宇清教授（社会化协同计算、数据安全与隐私保护、自然语言处理等）
- 学士论文：设计并实现隐私策略管理的中间件系统

工作经验

01/2020 – 09/2020 **研究实习生**，美国西北太平洋国家实验室（PNNL），美国华盛顿州
- 导师: 李昂博士
- 导师研究领域：硬件加速器、GPU、FPGA、计算机全栈设计–硬件/系统/库/应用

10/2018 – 12/2019 **访问研究助理**，罗彻斯特大学, 美国纽约州
- 导师：John Criswell教授，LLVM创始骨干
- 导师研究领域：安全操作系统、编译器、程序分析

05/2018 – 08/2018 **Google Summer of Code 学生**，与 The Honeynet Project 研究机构合作
- 导师：Tamas K Lengyel博士
- 导师研究领域：系统底层安全（入侵检测、操作系统、虚拟化、病毒木马分析）

08/2015 – 05/2018 **助教**, 威廉玛丽学院
- 主讲教师: Dr. Tim Davis, Dr. Xu Liu, Debbie Noonan, Dr. Amiya Bhattacharya
- 授课内容：编程语言原理、计算机动画、编程入门、代码与数据安全

主导/合作项目

12/2019 - 08/2021

项目（主导）： 利用内存标签在进程内部实现字节粒度的权限隔离与指针防护

背景： 进程地址空间内部的隔离一直是系统安全的重要保障机制，但现有的内存隔离技术（如 CPU Rings，Intel SGX/MPK，ARM TrustZone/MPU）均是大粒度的隔离，无法满足程序内部权限隔离对内存细粒度访问控制的需求；更重要的，应用这些技术对程序改动大，既耗费人力又有引入新风险的可能。

概述： 本工作扩展了现有的硬件处理器，以增加对内存地址空间细粒度访问控制的底层支持，提升计算机系统安全；此外，我们提供了简单易用的编译工具链，传统程序无须大量改动就可以获得保护。具体的，我们将传统的冯诺依曼体系结构扩展成为一类特定的标签体系结构 (Tagged Architecture)，使得每个内存字节都被赋予一个特定的标记，计算机系统进而可以利用该标记将内存空间分割成为更多不同的安全域。为兼容传统的应用程序，该工作对计算机的处理器、编译器、链接/加载器、操作系统均有相应的创新设计与适配工作，有效规避了对应用程序的移植成本，提升了移植的可靠性。

挑战: **a**）如何自下而上对计算机所有系统层进行综合设计，设计空间的变量涉足处理器体系结构、编译工具链、操作系统、用户程序。**b**）如何无缝兼容现有应用程序，把新设计的系统功能的向后兼容能力发挥到最大效果。**c**）在切实可行的基础上，如何在广阔的系统设计空间中寻找最优设计点，以减少整体系统的复杂度，提高整体的运行效率。

指导老师：李昂博士，John Criswell教授，李群教授

收获技能： 权限分离 (privilege separation)；影子内存 (shadow memory)；标签处理器架构 (tagged architecture)；MIPS R4000 流水线、协处理器、缓存的实现；Gem5; QEMU; FreeBSD; LLVM CodeGen; 链接器、加载器；ELF 文件；C 语言安全；硬件描述语言 Bluespec System Verilog.

02/2019 – 11/2019

项目（主导）： 借助硬件权能防止恶意操作系统内核对应用程序的攻击（基于 SAFECode, SVA, 和 CHERI）

背景： 传统计算环境中，操作系统内核一旦被入侵，往往会导致该系统内的所有应用程序被恶意控制。因此，如何在操作系统内核不安全的情况下保护应用程序，是重要课题。

概述： 基于最新的硬件权能处理器 CHERI, 我们对编译器和操作系统进行了协同扩展设计，加强内核不同功能之间的隔离，从而提高了用 C 语言编写的操作系统内核的安全性。具体的，基于之前的安全虚拟架构 (SVA)，我们将 FreeBSD 内核分割为两大区间：一个是可信的、较小的内核空间，余下部分作为不可信的、较大的内核空间。通过对应用程序的内存管理，以及对软硬件接口的严格访问控制，可信的内核空间可以直接给应用程序提供防护，而不可信的内核空间无法恶意控制应用程序。

挑战： **a**）如何界定可信小内核与不可信大内核的界限，以实现最小的可信内核；**b**）如何利用 CHERI 限制可信小内核与不可信内核之间的通信，在维持正常功能的前提下维持可信小内核的安全属性；**c**）鉴于 SVA 的设计思路，如何在 CHERI 平台上阻断不可信内核程序对上层应用程序的攻击。

指导老师：John Criswell教授，李群教授

收获技能： C 语言安全; 胖指针 (fat pointer); 处理器权能 (architectural capability); 控制流完整性 (control flow integrity); 内核权限分离 (kernel privilege separation); FreeBSD 内核 (context switch, exceptions, segmentation, thread local storage, etc.); LLVM/Clang; C.

03/2019 – 12/2019

项目（合作）： Silhouette: 利用 ARM 非特权指令集在嵌入式系统中实现高效的影子栈

背景： 当前多数嵌入式系统中的应用程序运行在内核态。这对程序的内存安全是极为不利的（如栈溢出、ROP/JOP 攻击）。而在内核中创建隔离空间将有效提高内存访问的安全性。

概述： 我们借助 ARM 处理器的非特权指令子集 (unprivileged instructions)，基于编译器开发了一种在内核中创建新隔离空间的方法。具体的，编译器会使用非特权的 load 和 store 指令在同一个地址空间中创建出两个不同的安全域。最终效果是，内核态下 “拥有特权” 的用户代码，无法访问编译工具链指定的受保护的内存区域。作为示例，我们实现了一个安全的应用程序栈，结合后向/前向控制流完整性 (Backward/Forward CFI) 的保护，达到防止有漏洞的应用代码遭受 ROP/JOP 等控制流攻击。

我的任务是负责操作系统 FreeRTOS 方面的开发： **a)** 通过对 ARM MPU 和链接脚本的设置，配置内核空间中的内存分配和权限分配；**b)** 在编译器生成代码的阶段，扫描即将生成（或已经生成）的二进制程序，搜寻出对于内存隔离机制有威胁的特权指令并报警；**c)** 在 FreeRTOS 中开启并配置 MPU；**d)** 在 FreeRTOS 中创建多个进程，实现多进程并发.

指导老师：John Criswell 教授，Robert J Walls 教授

收获技能： ARM assembly；ARM Cortex-M 内存管理 (MPU)；FreeRTOS 内核 (context switch. exceptions, process management, etc.); 程序栈安全（影子栈）；控制流完整性; LLVM MIR (prologue, epilogue).

10/2018 – 02/2019

项目（主导）： 将新版本的 FreeBSD 移植到安全虚拟架构 Secure Virtual Architecture(SVA)

背景: SVA是 LLVM 中间语言（LLVM IR) 设计的延续项目，主要致力于增加 LLVM IR 对操作系统行为的描述能力，用以构建一个基于编译器技术的虚拟运行平台。然而，随着操作系统（如 FreeBSD）的更新，对于 SVA 的兼容性出现了问题，使得新版本操作系统无法运行在 SVA 上，这也反过来限制了 SVA 无法运行新版本操作系统所支持的新硬件。因此，需要将新版本的操作系统移植到 SVA 上来解决问题。

概述: 本工作将新版本 FreeBSD 移植到 SVA 架构上，主要的挑战有：**a)** 定位新系统与 SVA 不兼容问题的根源；**b)** 增强 SVA 对于操作系统中线程本地存储 (TLS) 的支持，譬如在进程切换等过程中的寄存器备份与恢复时，增加 TLS 相关的操作；**c)** 在操作系统内核和程序库中开启 TLS 相关功能（之前移植到 SVA 的系统中此功能维持关闭）；**d)** 把 FreeBSD 新旧版本的差异重新生成补丁，使用 Git 等工具，以已经移植到 SVA 的旧版 FreeBSD 为基础，采用半人工的方式重新打上 FreeBSD 的新补丁，过程中人工确认、修正所有的内核更新。

指导老师：John Criswell教授

收获技能: FreeBSD 内核（Context switch; interrupt handling）；进程本地存储（thread local storage）；分段内存管理（segmentation）；LLVM intrinsics.

05/2018 – 08/2018

项目（主导）： 扩展TinyVMI – 轻量操作系统中实现的虚拟机内省

背景： 商用操作系统的安全漏洞层出不穷，是黑客攻击的重要目标之一。如何检测操作系统是否受到攻击，以及攻击之后如何有效应对，是重要的研究课题。而操作系统一旦被攻陷，任何运行在操作系统内部的安全软件都不再可信。因此，如果能在操作系统外部对操作系统内部的运行状态进行监测，有利于对攻击行为的监测。虚拟机内省技术便有助于实现这一目标。

概述： TinyVMI 是本人自硕士期间开始的一个研究项目，它实现了在一个轻量操作系统中对商用操作系统的内存监测。此次借助 Goolge Summer of Code 活动，与 The Honeynet Project 研究机构合作，对 TinyVMI 的检测功能进行了全面的扩展。扩展的内容有：增加了对硬件事件的支持；增加了对 Windows 和 Linux 内核配置文件的支持。扩展之后，TinyVMI 可以更有效的对商用操作系统的内存进行检测。该项目已经开源 https://tinyvmi.github.io.
挑战： **a)** 理清微型 OS 中的系统库、应用库的依赖关系并移植或者手工开发新库取代；**b)** 微型 OS 中如何与 Xen 硬件事件接口对接，并实现对事件的快速响应。
指导老师： Tamas K Lengyel 博士，资深安全研究员
收获技能： Xen 虚拟机监控器; 硬件事件; 虚拟机内省；程序库移植；交叉编译。

05/2016 – 04/2017

项目（主导）： 通过 Docker 分层文件系统实现边缘服务器进程的快速迁移
背景： 不同于服务器集中分布的云计算环境，边缘计算服务器的地理分布更加广泛，相互通信成本高、效率低，对进程实时迁移提出了新的挑战。
概述： 为提高边缘计算服务器之间的进程迁移效率，利用 Docker 容器 (container) 中对文件系统的分层管理原理 (layered storage)，在进程运行时向目标服务器迁移多个文件镜像层，而在进程停止，最后迁移时，仅仅发送较小的最顶层的容器层，从而把宕机时间减少了 56% ∼ 80%。代码已经公开https://github.com/tupipa/p.haul.
指导老师：李群教授
收获技能: Union Mount File System(AUFS)，进程迁移，CRIU, Go 语言, Python 语言。

05/2014 – 06/2015

项目（主导子项）： 面向系统安全的 CPU 和 OS 协同设计研究
背景： 硬件安全是整个计算机系统安全的基石，而近年来出现的硬件漏洞，更印证了研究安全硬件对计算机系统的重要性。将安全硬件与运行之上的基础软件协同设计以提供高安全的计算机系统是未来计算环境的重要革新方向。
概述： 致力于中央处理器 (CPU) 与操作系统 (OS) 的协同设计，以提升计算机整个系统的安全性。该项目从四个方面对 CPU 和 OS 的进行了研发：a) CPU 微码 (micro-codes) 的防护；b) 计算机可信启动 (trusted booting)；c) 动态完整性监控与检测；d) 虚拟机内省 (virtual machine introspection).
我的任务： **(硕士论文)** TinyVMI: 在一个轻量的操作系统中利用虚拟机内省对内存进行安全监测。设计并实现了在 Xen Mini-OS 中对传统虚拟机执行虚拟机内省的各种操作以监控内存变化。在安全方面，内存监控 VM 与被监控的 VM 通过 Xen Security Module(XSM) 执行强制访问控制，被监控 VM 可以执行任意操作而不影响监控信息的可信性；轻量 VM 可以检测到其他 VM 内核中静态数据遭到的篡改攻击，比如中断向量表 (IDT) 等。在性能上，轻量 VM 为单进程的 VM，占用内存极少 (<32MB)，监控功耗小。
指导老师：杨秋松 研究员
收获技能： 内存管理；内存完整性检测；虚拟机内省；Xen Mini-OS；FLASK 安全规则。

07/2013 – 05/2014

项目（合作）： 综合软硬件漏洞的攻击方法研究
背景: 随着借助处理器缓存漏洞、微码漏洞的出现，基于硬件、软件漏洞的综合计算机攻击方法不断出现，而且与基于纯软件漏洞的攻击更难有效防护，因此有必要对综合软硬件漏洞的攻击方法进行实战研究。

概述: 本项目致力于重现业界知名的硬件、底层系统漏洞，并研究漏洞对计算机系统安全的影响。譬如，针对以下四类漏洞进行了深入调研并复现：**a)** CPU 微码更新漏洞；**b)** CPU 缓存漏洞；**c)** CPU 系统管理模式 (SMM) 攻击方法；**d)** CPU 硬件虚拟机扩展的攻击方法。
我的任务: 调研并复现 CPU 缓存漏洞和硬件虚拟机扩展的攻击方法。
具体的，成功构建了两个攻击场景：**a)** 利用 CPU 缓存的漏洞实现了越权读写 SMRAM，替换了 SMM 模式中的 SMI Handler。**b)** 利用 x86 CPU 硬件扩展，复现了基于 NewBluePill((源自 Invisible Things Lab)) 的系统攻击，成功将 Native 运行中的 Windows 操作系统在线托起为一个虚拟机，而无须重启。
指导老师：杨秋松 研究员
收获技能: 系统管理模式（System Management Mode）；CPU 缓存管理（X86 MTRR); 硬件虚拟机扩展（Hardware Virtual Machine）；**必备语言**: IA32 汇编, C 语言.

11/2011 – 06/2012

项目（合作）：社交网络中隐私策略的个性化定义与自动验证
概述: 制定社交网络的隐私定义机制，允许用户自定义自己数据的隐私策略，且系统能自动检测用户规则的冲突规则，并提出合理的解决方案。
我的任务: **(学士论文)** 设计并实现隐私策略管理的中间件系统。
具体的，将用户定义的隐私策略转化为 Prolog 逻辑语句，并借助逻辑推理引擎 (tuProlog engine）对用户隐私策略建模，然后自动检测用户自定义策略中的冲突。用户可以根据系统的建议改进自己的隐私策略，最终消除冲突。
指导老师：：孙宇清教授
收获技能: 访问控制模型 (RBAC, ABAC); 一阶逻辑语言编程; **语言**: Java (>20KLoC)，Prolog.

学术论文

1. Lele Ma. One layer for all: Efficient system security monitoring for edge servers. In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2021.

2. (CCF A 类) Jie Zhou, Yufei Du, Zhuojia Shen, **Lele Ma**, John Criswell, and Robert J Walls. Silhouette: Efficient Protected Shadow Stacks for Embedded Systems. In *Proceedings of the 29th USENIX Security Symposium*, Security '20, pages 1219–1236, Boston, MA, 2020. USENIX Association.

3. (CCF A 类) Zeyi Tao, Qi Xia, Zijiang Hao, Cheng Li, **Lele Ma**, Shanhe Yi, and Qun Li. A Survey of Virtual Machine Management in Edge Computing. *Proceedings of the IEEE*, 107(8):1482–1499, 2019.

4. (CCF A 类) **Lele Ma**, Shanhe Yi, Nancy Carter, and Qun Li. Efficient Live Migration of Edge Services Leveraging Container Layered Storage. *IEEE Transactions on Mobile Computing*, 18(9):2020–2033, 2019.

5. (新兴会议) **Lele Ma**, Yantao Li, and Gang Zhou. A Video Optimization Framework for Tracking Teachers in the Classroom. In *ACSIC Symposium on Frontiers in Computing (SOFC, 北美计算机华人学者年会)*. ACSIC, 2018.

6. (新兴会议) **Lele Ma**, Shanhe Yi, and Qun Li. Efficient Service Handoff Across Edge Servers via Docker Container Migration. In *IEEE/ACM Symposium on Edge Computing (SEC, 边缘计算前言会议)*. IEEE, 2017.

7. (CCF C 类) **马乐乐**，岳晓萌，王玉庆，杨秋松. 基于轻量操作系统的虚拟机内省与内存安全监测. 计算机应用, 35(6):1555–1559, 2015.

8. (CCF B 类) 王媛，孙宇清，**马乐乐**. 面向社会网络的个性化隐私策略定义与实施. 通信学报, 33(Z1):239, 2012.

教学经验

2018 秋季	嘉宾授课	(CSCI 680) 高级计算机系统 (Advanced Topics in Computer Systems)
2018 春季	助教	(CSCI 312) 编程语言原理 (Principles of Programming Languages)
2017 秋季	助教	(CSCI 420-2) 计算机动画 (Computer Animation)
2017 春季	助教	(CSCI 312) 编程语言原理 (Principles of Programming Languages)
2016 秋季	助教	(CSCI 312) 编程语言原理 (Principles of Programming Languages)
2016 春季	助教	(CSCI 141) 编程入门 (Computational Problem Solving)
2015 秋季	助教	(CSCI 420) 代码与数据安全 (Data and Code Security)

获奖情况

2018-2019	Research Assistant Scholarship, 罗彻斯特大学
2017-2018	Teaching Assistantship, 威廉玛丽学院
10/2017	Student Travel Grant，美国自然科学基金 (NSF)
09/2017	Conference Funding Award，威廉玛丽学院
09/2017	GSA Conference Travel Award, 威廉玛丽学院
2016-2017	Teaching Assistantship, 威廉玛丽学院
2015-2016	Teaching Assistantship, 威廉玛丽学院
2014-2015	研究生研究助理奖学金, 中国科学院大学
2013-2014	研究生研究助理奖学金, 中国科学院大学
2012-2013	研究生研究助理奖学金, 中国科学院大学
12/2011	山东大学优秀学生奖学金
08/2010	全国田径锦标赛优秀志愿者
06/2010	山东大学优秀共青团员
11/2009	山东大学优秀学生奖学金

Lele Ma

Last Updated on 01/2022

EMAIL: lma03@wm.edu PHONE (CHINA): +86 135 5208 4690
TIME ZONE: China Standard Time

INTERESTS & SPECIALTIES

General Interests	- **OS Kernel, System Security, Memory Safety, Privilege Separation;** - **System Support for Cloud/Edge computing.**
Specialties	- Hardware-software co-design for security; - Memory safety in operating system kernel; - Intra-address space isolation; - Virtual machine introspection; - Docker container migration; - Programming languages: • Professional in C, C++, Java; • Familiar with Go, Python, Bash; • Amateur level on Coq, Haskell, Prolog.
Exemplar Project	- TinyVMI (https://github.com/tinyvmi/tinyvmi/). - See More at Research Projects and Publications

EDUCATION

08/2015 – 01/2022	**Ph.D. in** COMPUTER SCIENCE College of William & Mary, Virginia, USA - Advisor: Prof. Qun Li - Thesis: Revisiting Isolation Mechanisms for System Security and Efficiency in the Era of Internet of Things
09/2012 – 07/2015	**Master Degree in** COMPUTER SCIENCE Institute of Software, Chinese Academy of Sciences, Beijing, China - Advisor: Prof. Qiusong YANG - Thesis: Design and Implementation of a Lightweight VM Monitor Based on Virtual Machine Introspection
09/2008 – 06/2012	**Bachelor Degree in** COMPUTER SCIENCE Shandong University, Jinan, Shandong Province, China - Advisor: Prof. Yunqing Sun - Thesis: Middleware System Design and Implementation for Privacy Policy Management

WORK EXPERIENCE

01/2020 – 09/2020	**Intern at PNNL** Pacific Northwest National Laboratory, Richland, WA, USA Mentor: Dr. Ang Li
10/2018 – 12/2019	**Visiting Research Assistant** University of Rochester, Rochester, NY, USA Advisor: Prof. John CRISWELL
05/2018 – 08/2018	**Google Summer of Code Student** Participated with The Honeynet Project - Mentor: Dr. Tamas K LENGYEL
08/2015 – 05/2018	**Teaching Assistant** The College of William and Mary, Virginia, USA - Lectured by: Dr. Tim DAVIS, Dr. Xu LIU, Debbie NOONAN, and Dr. Amiya Bhattacharya

Research Projects

12/2019 – 08/2021

Project: Fine-grained privilege separation with ownership for every byte in memory
Project Description: In order to separate the memory into different security domains, we extend the traditional Von Neumann architecture with features of tagged architectures: binding every memory byte with an ownership identity. Then, with the enhancement of the LLVM-based toolchain (compiler, linker, loader, etc.), ownership is assigned to every data word and every instruction. Finally, the whole memory is divided into different security domains identified by their ownership. Our work provides better flexibility, compatibility, and stronger memory protection than traditional isolation mechanisms such as Intel SGX/MPK, ARM TrustZone/MPU, etc.
Key challenges: **a)** How to implement the ownership for every byte in the memory by revising system stack from bottom up, including the hardware processor, the operating system, the compiling toolchain, and user programs; **b)** How to seamlessly support the legacy software while retain minimal human refactoring efforts, aiming for the backward compatibility with most of the legacy programs; **c)** Where to choose a point in the design space of software-hardware co-design, in order to reduce the overall complexity of the system and improve the overall efficiency, while keeping memory protected.
Skills: privilege separation; shadow memory; tagged architecture; Implementation of MIPS R4000 (pipeline, coprocessors, caches); Gem5; QEMU; FreeBSD; LLVM (CodeGen); linker and loader (LLD, ELF format); language safety (in C); Bluespec System Verilog.

02/2019 – 11/2019

Project: Protecting applications from malicious kernel with hardware capability support (based on SAFECode, SVA, and CHERI)
Project Description: With the state-of-the-art hardware support in CHERI, we explore the new design space for the co-design of the compiler and the operating system, to bring memory safety to the OS kernel written in the C language, reducing the effects of kernel bugs and vulnerabilities. Based on previous work on SVA, we retrofit the FreeBSD kernel where two compartments are build: one minimal trusted compartment with a small amount of kernel code and a bigger untrusted compartment contains all the rest of the kernel code. The small compartment of kernel provides memory safety guarantees protecting applications against the malicious kernel portion by taking control all the application's memory states and carefully constrained low-level hardware-software interfaces.
Key challenges include the definition of a minimal secure portion of kernel, how to isolate trusted and untrusted portions of the kernel and limit their interactions using sealing mechanisms in CHERI, and how to protect applications from untrusted portion of the kernel based on SVA.
Skills: language safety; fat pointers; architectural capability; control flow integrity; kernel privilege separation; FreeBSD kernel(context switch, exceptions, segmentation, thread local storage, etc.); LLVM/Clang; C.

03/2019 – 12/2019

Project: Silhouette: Efficient Protected Shadow Stacks for Embedded Systems
Project Description: Most embedded devices run applications in privileged mode which sacrifices memory safety (stack overflow, ROP/JOP attacks). With the latest hardware feature in ARM processors, we explore new efficient ways for intra-address space isolation. Unprivileged load and store instructions are leveraged by our compiler to automatically isolate different regions inside the same address space. Conceptually, we build a region pf memory where the privileged user on the system can not access. The region is used as a shadow stack to enforce backward control flow integrity(CFI) for the privileged application. In addition, we also build label-based CFI to protect function pointers by patching a dummy label to each function header and checking the label before an indirect call.
My Tasks: a) Configure the MPU and linker scripts to allocate and control permissions for different memory regions in the address space; **b)** In the code generation stage, scan the emitted native code and ensure that no privileged instructions exists in the application's code that can be used to bypass the shadow stack protection; **c)** Enable MPU in FreeRTOS; **d)** Compose concurrent multiple processes under FreeRTOS.
Skills: language safety; control flow integrity; ARM Cortex-M MPU; FreeRTOS (context switch. exceptions, process management, etc.); LLVM MIR (prologue, epilogue).

10/2018 – 02/2019

Project: Porting Newer FreeBSD to Secure Virtual Architecture (SVA)

Project Description: Original Secure Virtual Architecture only supports an old version of FreeBSD (9.0), but this version of FreeBSD lacks support for modern hardware. In order to be compatible with the latest hardware, SVA needs a newer version of FreeBSD (such as 9.3) to be ported. Key challenges include a) enhancing SVA with thread local storage (TLS) support; b) enabling TLS support in the kernel/library code where were disabled previously by SVA; c) with help of git, re-apply the SVA patch (for FreeBSD 9.0) to FreeBSD 9.3 and validate the correctness of every change.
Skills: Context switch; interrupt handling; thread local storage (TLS); segmentation; LLVM intrinsics; control flow integrity; C.

05/2018 – 08/2018

Project: Enhancing TinyVMI – Virtual Machine Introspection in a minimal OS
Description: Google Summer of Code 2018 with The Honeynet Project.
TinyVMI is our virtual machine introspection library implemented in a minimal OS, Xen Mini-OS. In this summer, TinyVMI is enhanced by adding hardware event support and kernel profile parsing support. After enhancement, Mini-OS in a guest VM can introspect the runtime status of other guest VMs based on pre-registered hardware events. This project was developed in collaborate with Dr. Tamas K Lengyel. VMI operations are ported from LibVMI. **The challenge** is how to interact with interfaces of Xen hypervisor and resolving library dependencies in the Mini-OS. TinyVMI is now open-sourced.
Skills: Xen hypervisor; hardware events; virtual machine introspection; library porting; cross-compiling.

05/2016 – 04/2017

Project: Service Handoff across Edge Servers via Docker Container Migration
Description: Enable seamless service handoff across edge servers by fast migrating the Docker container while mobile clients is moving from current edge server towards another one. Layered storage was leveraged to share common layers between hosts before the actual live migration starts. Migration time is reduced by $56\% \sim 80\%$ compared to the state-of-the-art. Code is now open-sourced.
Skills: union mount file system(AUFS); process migration; **languages**: Go, Python.

05/2014 – 06/2015

Project: CPU and OS Integrated Security Research
Project Tasks: Integrated protection for the system, mainly utilizing 4 kinds of techniques: CPU micro-codes enhanced protection, trusted booting, dynamic integrity measurments, and virtual machine introspection.
My Task: (**Master Thesis**) Kernel Integrity Checking via Virtual Machine Introspection Based on Mini-OS—the tiny operating system in Xen source tree.
Design and implement the integrity checking agent based on a Mini-OS domain on Xen platform. The agent is well isolated through the Mandatory Access Control by XSM; it's lightweight with much less performance overhead; it's clean-state with little expansion of the TCB of the system. It can detect malicious behaviors that attempt to tamper with the critical codes or data of the runtime kernel, like the IDT table, keyboard handler, etc.
Skills: runtime integrity measurements; virtual machine introspection; para-virtualization; Xen and Mini-OS; **language**: C.

07/2013 – 05/2014

Project: Design of Hardware & Software Integrated Security Attack Scenarios
Project Tasks: Reproduce the hardware bugs and evaluate their hazards to operating systems under certain scenarios. Mainly contain 4 bugs to design the attack scenarios: CPU micro-codes updating bugs, CPU cache vulnerability, System Management Mode attacks, and Hardware Virtual Machine attacks.
My Task: Research on CPU cache vulnerability & Hardware Virtual Machine Attacks.
Mainly doing experiments on two scenarios: experiment that exploits the vulnerability of CPU cache to attack SMM codes stored in SMRAM; experiment that reproduces attacks on the HVM platform via NewBluePill (from Invisible Things Lab)
Skills: System Management Mode; CPU cache control via MTRR; Hardware Virtual Machine; **language**: IA32 assembly, C.

11/2011 – 06/2012

Project: Personalized Privacy Policy Definition and Verification
Prject Tasks: Explore the mechanisms of how the privacy policies can be personally defined by users, and how to check and resolve the conflicts among the personalized policies.

My Tasks: (**Bachelor Thesis**) Design and Implementation of Privacy Policy Management Middleware System.
The system allows users to define personalized privacy policies on their online private resources. The system uses tuProlog logic engine to model privacy policies and check the potential conflicts among the personalized policies. Conflicts can be detected via the logic engine and can be resolved by users' choice.
Skills: Access Control (RBAC, ABAC); First-order logic programming; **language**: Prolog, Java (>20KLoC).

Publications

1. Lele Ma. One layer for all: Efficient system security monitoring for edge servers. In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2021.
2. Jie Zhou, Yufei Du, Zhuojia Shen, **Lele Ma**, John Criswell, and Robert J Walls. Silhouette: Efficient Protected Shadow Stacks for Embedded Systems. In *Proceedings of the 29th USENIX Security Symposium*, Security '20, pages 1219–1236, Boston, MA, 2020. USENIX Association.
3. Zeyi Tao, Qi Xia, Zijiang Hao, Cheng Li, **Lele Ma**, Shanhe Yi, and Qun Li. A Survey of Virtual Machine Management in Edge Computing. *Proceedings of the IEEE*, 107(8):1482–1499, 2019.
4. **Lele Ma**, Shanhe Yi, Nancy Carter, and Qun Li. Efficient Live Migration of Edge Services Leveraging Container Layered Storage. *IEEE Transactions on Mobile Computing*, 18(9):2020–2033, 2019.
5. **Lele Ma**, Yantao Li, and Gang Zhou. A Video Optimization Framework for Tracking Teachers in the Classroom. In *ACSIC Symposium on Frontiers in Computing (SOFC)*. ACSIC, 2018.
6. **Lele Ma**, Shanhe Yi, and Qun Li. Efficient Service Handoff Across Edge Servers via Docker Container Migration. In *IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2017.
7. **Lele Ma**, Xiaomeng Yue, Yuqing Wang, and Qiusong Yang. Virtual Machine Introspection and Memory Security Monitoring Based on Light-weight Operating System. *Journal of Computer Applications (in Chinese)*, 2015.
8. Yuan Wang, Yuqing Sun, and **Lele Ma**. Specification and Enforcement of Personalized Privacy Policy for Social Network. *Journal of China Institute of Communications (in Chinese)*, 33, 2012.

Teaching Experience

09/2018	Guest Lecturer	CSCI 680 Advanced Topics in Computer Systems
Spring 2018	Teaching Assistant (Grader)	(CSCI 312) Principles of Programming Languages
Fall 2017	Teaching Assistant (Grader)	(CSCI 420-2) Computer Animation
Spring 2017	Teaching Assistant (Grader)	(CSCI 312) Principles of Programming Languages
Fall 2016	Teaching Assistant (Grader)	(CSCI 312) Principles of Programming Languages
Spring 2016	Teaching Assistant (Grader)	(CSCI 141) Computational Problem Solving
Fall 2015	Teaching Assistant (Grader)	(CSCI 420) Data and Code Security

Scholarships and Certificates

2018-2019	Research Assistant Scholarship, University of Rochester
2017-2018	Teaching Assistantship, College of William & Mary
10/2017	NSF Student Travel Grant
09/2017	Conference Funding Awards from Student Leadership Development
09/2017	GSA Conference Travel Award, College of William & Mary
Summer 2017	Research Assistant Scholarship, College of William & Mary
2015-2017	Teaching Assistantship, College of William & Mary
2012-2015	Research Assistantship, University of Chinese Academy of Sciences
12/2011	Shandong University Outstanding Student Scholarship
08/2010	Outstanding Volunteer of National Track and Field Championships
06/2010	Excellent League Member of Shandong University
11/2009	Shandong University Outstanding Student Scholarship