

USENIX Security 2009報告  
at 19回仮想化実装技術勉強会(vimpl)  
2009/Sep/11

須崎有康

# 傾向と対策

- 投稿数176 うち172(withdraw 1 and summary reject 3)
- 最終候補62 paper discussed in 1.5 days (all member attend)
- 26 accepted
- 半分は締め切りの直前3時間。このうち18本採択されている。1月の投稿は採択なし。
  
- 420人参加
  - Student grant 84 人のうち79人が取得
  
- 論文スライド資料はHPで公開
  - <http://www.usenix.org/events/sec09/index.html>
  
- Award
  - Compromising Electromagnetic Emanations of Wired and Wireless Keyboards (LASEC/EPFL )
  - Vanish: Increasing Data Privacy with Self-Destructing Data (U. Washington)
  
- Next
  - Washington 9-13/Aug 2010

# プログラム 1日目

- 9:00-10:30 **Keynote Address**
  - **Android: Securing a Mobile Platform from the Ground Up**
    - [Rich Cannings](#), *Android Security Leader, Google*
- 11:00-12:30 **Attacks on Privacy**
  - **Compromising Electromagnetic Emanations of Wired and Wireless Keyboards**
    - Martin Vuagnoux and Sylvain Pasini, *LASEC/EPFL*
  - **Peeping Tom in the Neighborhood: Keystroke Eavesdropping on Multi-User Systems**
    - Kehuan Zhang and XiaoFeng Wang, *Indiana University, Bloomington*
  - **A Practical Congestion Attack on Tor Using Long Paths**
    - Nathan S. Evans, *University of Denver*; Roger Dingledine, *The Tor Project*; Christian Grothoff, *University of Denver*
- 14:00–15:30 **Memory Safety**
  - **Baggy Bounds Checking: An Efficient and Backwards-Compatible Defense against Out-of-Bounds Errors**
    - Periklis Akritidis, *Computer Laboratory, University of Cambridge*; Manuel Costa and Miguel Castro, *Microsoft Research, Cambridge*; Steven Hand, *Computer Laboratory, University of Cambridge*
  - **Dynamic Test Generation to Find Integer Bugs in x86 Binary Linux Programs**
    - David Molnar, Xue Cong Li, and David A. Wagner, *University of California, Berkeley*
  - **NOZZLE: A Defense Against Heap-spraying Code Injection Attacks**
    - Paruj Ratanaworabhan, *Cornell University*; Benjamin Livshits and Benjamin Zorn, *Microsoft Research*
- 16:00–17:30 **Network Security**
  - **Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine**
    - Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, and Alexander G. Gray, *Georgia Tech*; Sven Krasser, *McAfee, Inc*
  - **Improving Tor using a TCP-over-DTLS Tunnel**
    - Joel Reardon, *Google Switzerland GmbH*; Ian Goldberg, *University of Waterloo*
  - **Locating Prefix Hijackers using LOCK**
    - Tongqing Qiu, *Georgia Tech*; Lusheng Ji, Dan Pei, and Jia Wang, *AT&T Labs—Research*; Jun (Jim) Xu, *Georgia Tech*; Hitesh Ballani, *Cornell University*

# プログラム 2日目

- 9:00-10:30 **JavaScript Security**
  - **GATEKEEPER: Mostly Static Enforcement of Security and Reliability Policies for JavaScript Code**
    - Salvatore Guarnieri, *University of Washington*; Benjamin Livshits, *Microsoft Research*
  - **Cross-Origin JavaScript Capability Leaks: Detection, Exploitation, and Defense**
    - Adam Barth, Joel Weinberger, and Dawn Song, *University of California, Berkeley*
  - **Memory Safety for Low-Level Software/Hardware Interactions**
    - John Criswell, *University of Illinois*; Nicolas Geoffray, *Université Pierre et Marie Curie, INRIA/Regal*; Vikram Adve, *University of Illinois*
- 11:00-12:30 **Radio**
  - **Physical-layer Identification of RFID Devices**
    - Boris Danev, *ETH Zürich, Switzerland*; Thomas S. Heydt-Benjamin, *IBM Zürich Research Laboratory, Switzerland*; Srdjan Čapkun, *ETH Zürich, Switzerland*
  - **CCCP: Secure Remote Storage for Computational RFIDs**
    - Mastrooreh Salajegheh, Shane Clark, Benjamin Ransford, and Kevin Fu, *University of Massachusetts Amherst*; Ari Juels, *RSA Laboratories, The Security Division of EMC*
  - **Jamming-resistant Broadcast Communication without Shared Keys**
    - Christina Pöpper, Mario Strasser, and Srdjan Čapkun, *ETH Zurich, Switzerland*
- 14:00–15:30 **Securing Web Apps**
  - **xBook: Redesigning Privacy Control in Social Networking Platforms**
    - Kapil Singh, *Georgia Institute of Technology*; Sumeer Bhola, *Google*; Wenke Lee, *Georgia Institute of Technology*
  - **Nemesis: Preventing Authentication & Access Control Vulnerabilities in Web Applications**
    - Michael Dalton and Christos Kozyrakis, *Stanford University*; Nikolai Zeldovich, *CSAIL, MIT*
  - **Static Enforcement of Web Application Integrity Through Strong Typing**
    - William Robertson and Giovanni Vigna, *University of California, Santa Barbara*
- 16:00–17:30 **Applied Crypto**
  - **Vanish: Increasing Data Privacy with Self-Destructing Data**
    - Roxana Geambasu, Tadayoshi Kohno, Amit A. Levy, and Henry M. Levy, *University of Washington*
  - **Efficient Data Structures for Tamper-Evident Logging**
    - Scott A. Crosby and Dan S. Wallach, *Rice University*
  - **VPriv: Protecting Privacy in Location-Based Vehicular Services**
    - Raluca Ada Popa and Hari Balakrishnan, *Massachusetts Institute of Technology*; Andrew J. Blumberg, *Stanford University*

# プログラム 3日目

- 9:00-10:30 **Malware Detection and Protection**
  - **Effective and Efficient Malware Detection at the End Host**
    - Clemens Kolbitsch and Paolo Milani Comparetti, *Secure Systems Lab, TU Vienna*; Christopher Kruegel, *University of California, Santa Barbara*; Engin Kirda, *Institute Eurecom, Sophia Antipolis*; Xiaoyong Zhou and XiaoFeng Wang, *Indiana University at Bloomington*
  - **Protecting Confidential Data on Personal Computers with Storage Capsules**
    - Kevin Borders, Eric Vander Weele, Billy Lau, and Atul Prakash, *University of Michigan*
  - **Return-Oriented Rootkits: Bypassing Kernel Code Integrity Protection Mechanisms**
    - Ralf Hund, Thorsten Holz, and Felix C. Freiling, *Laboratory for Dependable Distributed Systems, University of Mannheim, Germany*
- 11:00-12:30 **Browser Security**
  - **Crying Wolf: An Empirical Study of SSL Warning Effectiveness**
    - Joshua Sunshine, Serge Egelman, Hazim Almuhammedi, Neha Atri, and Lorrie Faith Cranor, *Carnegie Mellon University*
  - **The Multi-Principal OS Construction of the Gazelle Web Browser**
    - Helen J. Wang, *Microsoft Research*; Chris Grier, *University of Illinois at Urbana-Champaign*; Alex Moshchuk, *University of Washington*; Samuel T. King, *University of Illinois at Urbana-Champaign*; Piali Choudhury and Herman Venter, *Microsoft Research*
  - **The Multi-Principal OS Construction of the Gazelle Web Browser**
    - Helen J. Wang, *Microsoft Research*; Chris Grier, *University of Illinois at Urbana-Champaign*; Alex Moshchuk, *University of Washington*; Samuel T. King, *University of Illinois at Urbana-Champaign*; Piali Choudhury and Herman Venter, *Microsoft Research*

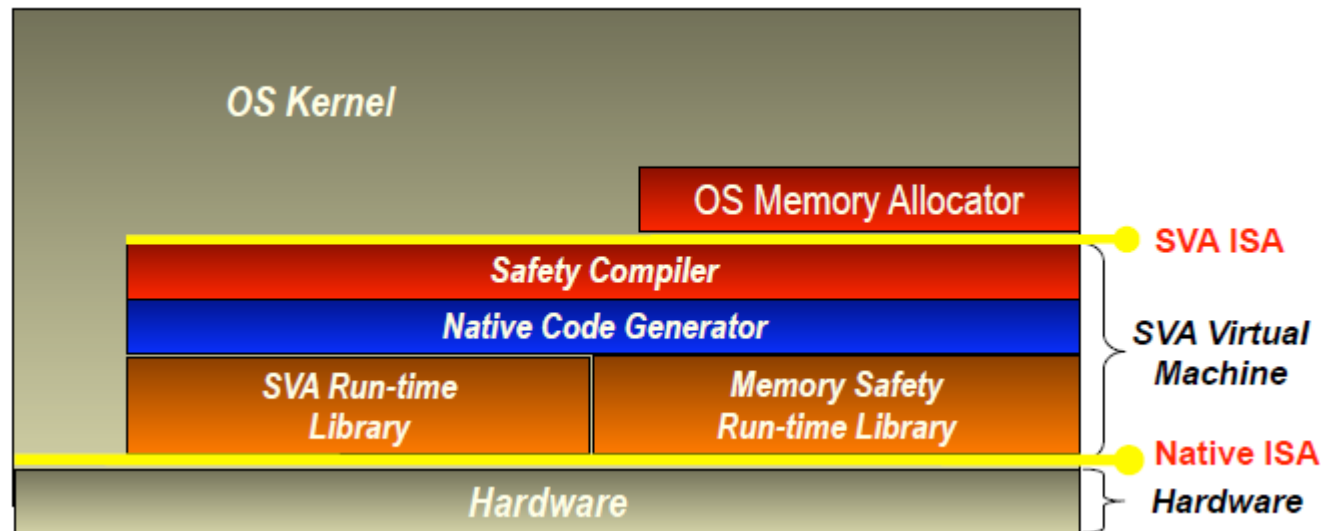
# Peeping Tom in the Neighborhood: Keystroke Eavesdropping on Multi-User Systems

– Kehuan Zhang and XiaoFeng Wang, Indiana University, Bloomington

- マルチユーザ環境での他人のキーボードストロークを検出する手法。
- Linuxの/proc/<pid>/stat情報から検出する。ESP (stack pointer)をトレースし入力文字を検出。
  - 他のUNIXではESPが見られない。

# Memory Safety for Low-Level Software/Hardware Interactions (1/2)

- John Criswell, University of Illinois; Nicolas Geoffray, Université Pierre et Marie Curie, INRIA/Regal; Vikram Adve, University of Illinois
- SVA (Secure Virtual Architecture [SOSP07].)でも起こるLow-Levelなメモリ安全性を保つための改良。
  - SVA はLLVMをベースとした Compiler-based Virtual Machine。Traditional memory safetyも保障。



# Memory Safety for Low-Level Software/Hardware Interactions (2/2)

- SVAではLanguage level safety (LLVMベース)と個々の処理のsafe executionを保証するが、他の処理によるlow levelな破壊まで保証できない。
  - プロセス状態保護
    - dangling pointerによる破壊
    - Self Modifying Code 自己書き換えコード(Linux のalternatives, ftraceなど)
      - 「IA-32 インテル アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル 下巻: システムプログラミングガイド」の10.6節 自己修正コード の項にはコードを書き換えたらcpuid命令(など)を発行してシリアルか操作を実行することが明記。
  - Memory Mapped I/O 破壊 (Linux 2.6のE1000e ドライババグ)
    - cmpexchg (compare-exchange命令)をMemory Mapped I/Oに適用したため。  
<http://lwn.net/Articles/304105/>
  - MMU でmap countが0になるバグ。
    - Linux 2.4 のmremap. BID9386 (<http://www.securityfocus.com/bid/9356>)
- SVAのsafety checkerの他にVerifierを加える。
  - SVAに命令追加。
    - 例1: sva\_begin\_alt, sva\_end\_alt 自己書き換えコードの領域指定
      - sva\_disable\_code, sva\_enable\_code自己書き換えコードの書換え不許可・許可
    - 例2: sva\_swap\_interger context swithした状態のIDで管理
  - 既存の SVA-Linuxを5,000 行修正。
  - 上記の問題を検出。

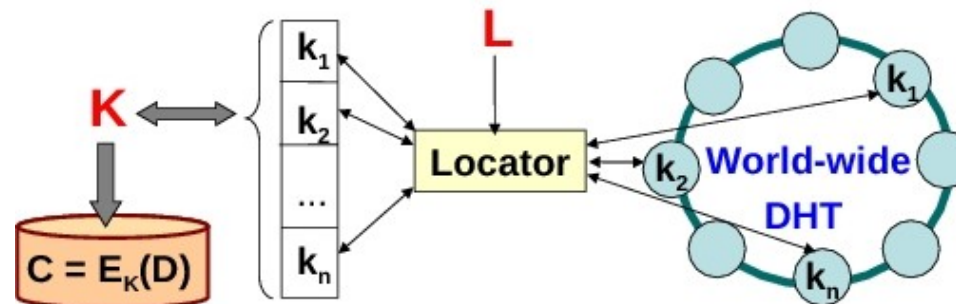
# SVA関連情報

- プロジェクトHP
  - <http://sva.cs.illinois.edu/>
- Secure Virtual Architecture: A Safe Execution Environment for Commodity Operating Systems (SOSP 2007)
  - <http://www.sosp2007.org/papers/sosp139-criswell.pdf>
- A virtual instruction set interface for operating system kernels (WIOSCA 2006)
  - <http://llvm.org/pubs/2006-06-18-WIOSCA-LLVAOS.html>
- Secure Virtual Architecture: Using LLVM to Provide Memory Safety to the Entire Software Stack
  - [http://llvm.org/devmtg/2008-08/Criswell\\_SVA.pdf](http://llvm.org/devmtg/2008-08/Criswell_SVA.pdf)

# Vanish: Increasing Data Privacy with Self-Destructing Data

– Roxana Geambasu, Tadayoshi Kohno, Amit A. Levy, and Henry M. Levy, University of Washington

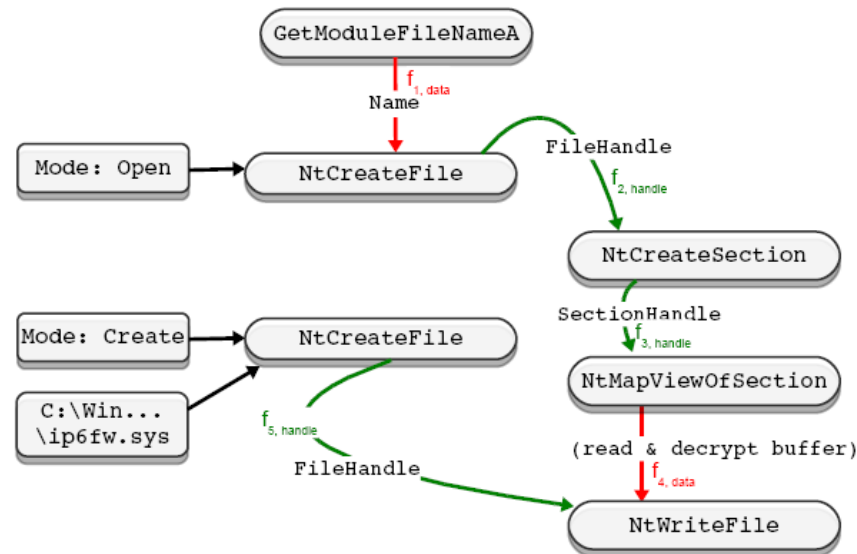
- Webメールなどで添付ファイルサーバに残るファイルを消す仕組み。
- DHTベースでdata keyとlocation keyがあり、location keyのサーバ上のデータが時間で無くなるため、データを再現できなくなる。



- Demo HPからテキストをvanish化できる。
  - <http://regina.cs.washington.edu/cgi-bin/vanishservice.py>
- FireFoxにpluginがあるが、Vuze DHT (Bittorrentベース) に依存しているので、職場で使うと問題あり。

# Effective and Efficient Malware Detection at the End Host (1/2)

- Clemens Kolbitsch, Paolo Milani Comparetti (TU Vienna), **Christopher Kruegel**(UCSB), **Engin Kirda** (Institute Eurecom, Sophia Antipolis); Xiaoyong Zhou and XiaoFeng Wang, Indiana (University at Bloomington)
- Windows の System Call の呼び出し関係を Behavior Graph として表現。単に呼び出し関係を表示するのではなく、引数の使われ方を解析し、Malware 独特の使われ方を学習。
  - 引数の解析をしないと False Positive を起こす。



- あらかじめ malware の behavior graph を Anubis(<http://anubis.iseclab.org/>)で収集。

# Effective and Efficient Malware Detection at the End Host (2/2)

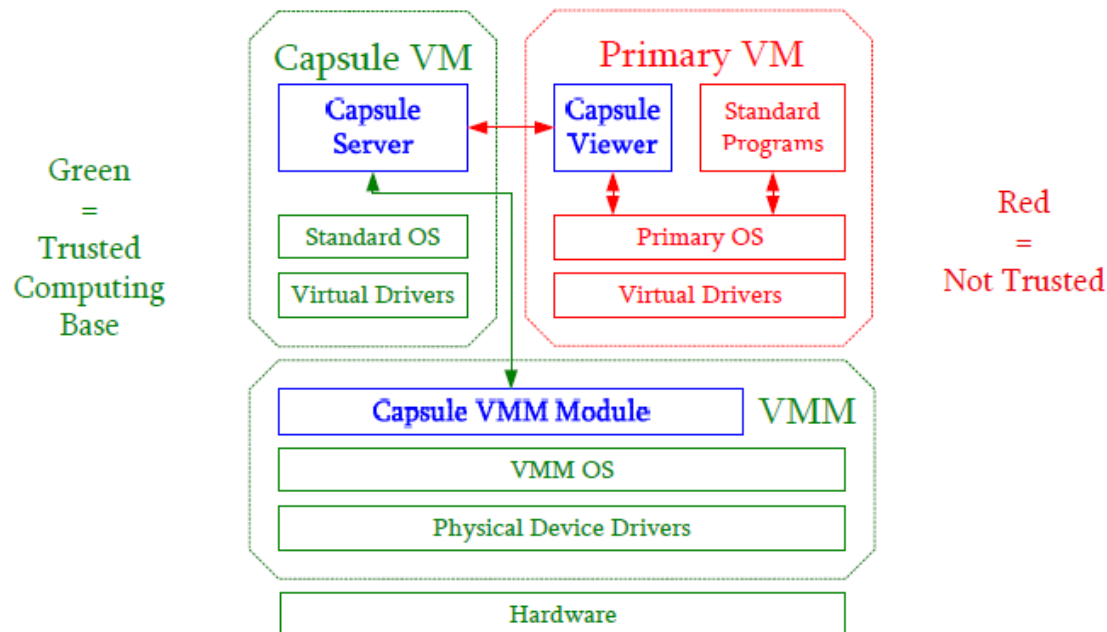
- 未知のプログラム実行する際に "dynamic program slicing" で表現したものを作成して、behavior graph 照らし合わせてること(matching behavior)で malware を動的に解析。



- Anubis(<http://anubis.iseclab.org/>)はバイナリファイルをアップロードするのみで解析する。ベースはTTAnalyzer(EICAR2006)?
- 同様のシステムとして virustotal と CWSandbox がある。
  - <http://www.virustotal.com/>
  - <http://www.cwsandbox.org/>

# Protecting Confidential Data on Personal Computers with Storage Capsules (1/2)

- Kevin Borders, Eric Vander Weele, Billy Lau, and Atul Prakash, University of Michigan
- 機密データをアクセスする際にVMMによるモード変換、デバイスアクセス制限、スナップショットによるロールバックを組み合わせることで情報漏洩・改竄を防止する。
- Secure モードでは
  - ネットワーク等の外部デバイス禁止
  - データアクセス前にスナップショット取得、アクセス後にロールバックを行う。
  - Covert Channel (VMのOver Commit memory、物理ディスクキャッシュ)をふさぐ。

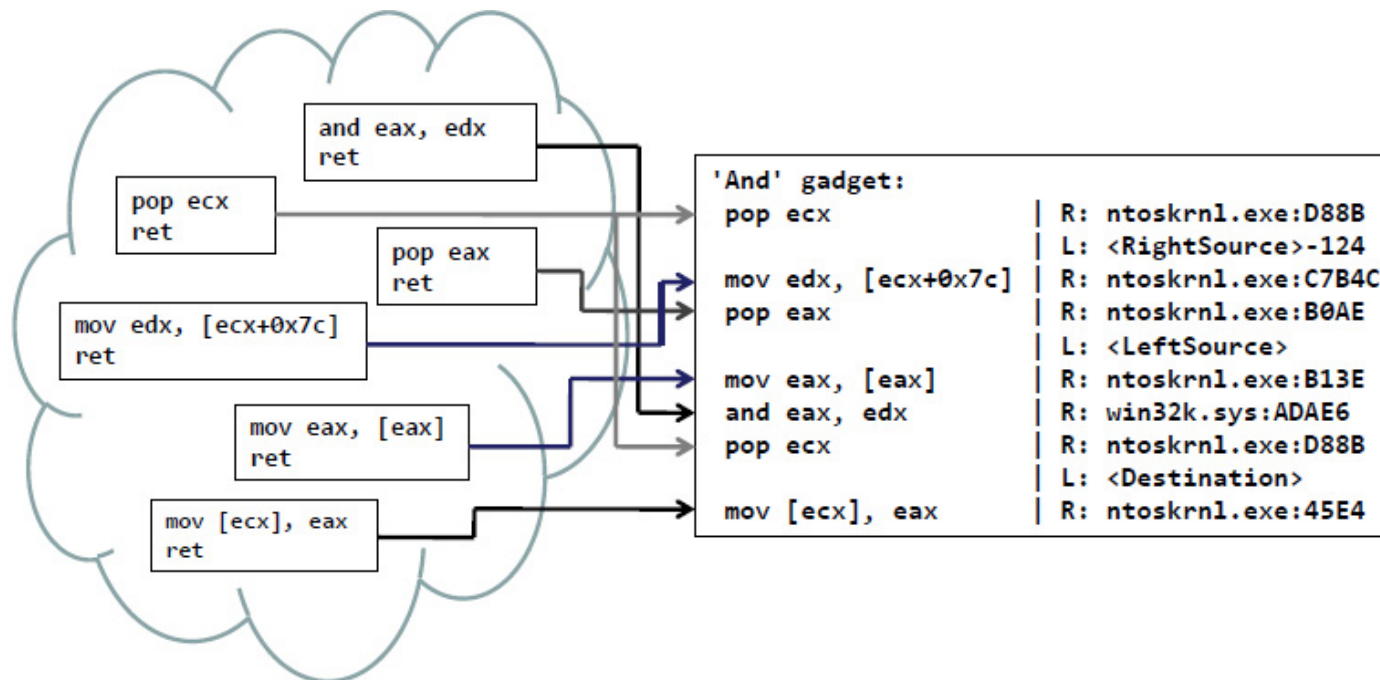


# Protecting Confidential Data on Personal Computers with Storage Capsules (2/2)

- 実装
  - VMMはVMWare。Snapshot revert後にデバイスをリセットできるにはVMWareのみ？
  - ユーザからは TrueCryptのように使えるらしい。
- 関連研究
  - VirtualATM
    - Windowsで同時に実行可能なプロセスを1つだけに制限する。
    - <http://www.authentium.com/developers/virtualATM.html>
  - P-MMAPS (Processor-Measured Application Protection Service)
    - Intel TXT (Trusted Execution Technology) を拡張し、アプリケーションを安全に実行する仕組み。
    - <http://www.intel.com/technology/itj/index.htm>

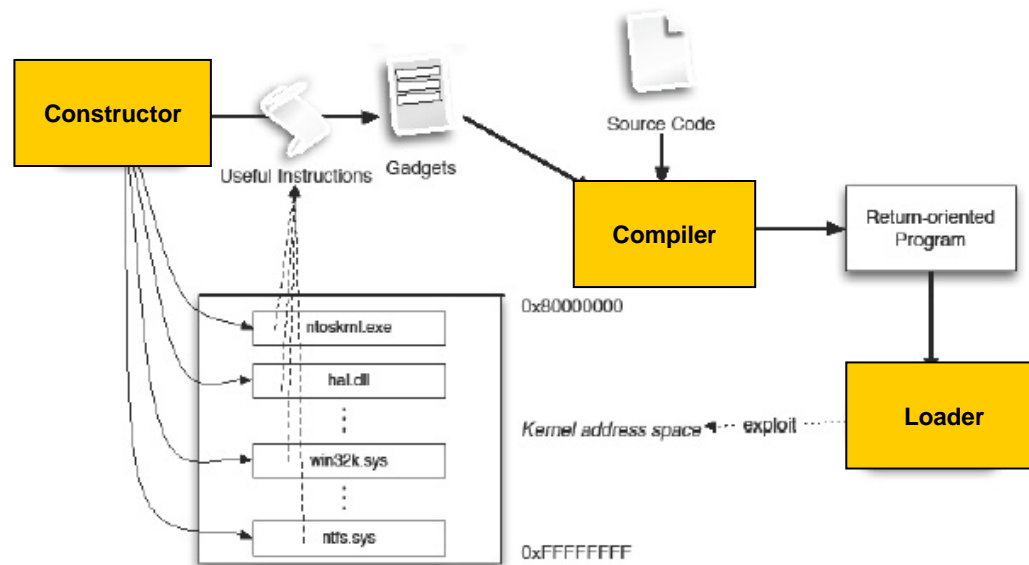
# Return-Oriented Rootkits: Bypassing Kernel Code Integrity Protection Mechanisms (1/2)

- Ralf Hund, Thorsten Holz, and Felix C. Freiling, Laboratory for Dependable Distributed Systems, University of Mannheim, Germany
- Return Oriented Programming ... 既存のコードから適当に命令を選んで悪意のあるコードを作ること。RET命令の前にあるコードが使いやすいため、Return Oriented と呼ばれる。
  - libcがよく使われたことからReturn to libc attack とも呼ばれてた[Shacham, CCS07]。



# Return-Oriented Rootkits: Bypassing Kernel Code Integrity Protection Mechanisms (2/2)

- カーネルコードから作成するツールを開発。
  - Constructor コードを収集ツール
  - Compiler コード生成ツール
  - Loader exploit ツール



- この攻撃はカーネルの改竄防止をする **SecVisor[ASPLOS08]**, **NICLE[RAID08]** で防げない。( **OverShadow[ASPLOS08]**, **Biton [BlackHat08]** でも防げない。)
- Reference
  - When Good Instructions Go Bad: Generalizing Return-Oriented Programming to RISC [CCS08, UCSD]

# Invite Talk: Top Ten Web Hacking Techniques 2008

Jeremiah Grossman, WhiteHat Security

- 違う機会に発表されたスライド。
  - <http://www.slideshare.net/jeremiahgrossman/top-ten-web-hacking-techniques-2008>

1. GIFAR = GIF + JAR
  - GIFファイルとしてアップロードし、JARとして動作する
2. Breaking Google Gears' Cross-Origin Communication Model  
Flash Parameter Injection
3. Safari Carpet Bomb
  - HTML内に簡単なスクリプトを仕込むだけで、任意のファイルをダウンロードフォルダに勝手に保存させることができる
4. Clickjacking / Videojacking
  - 通常のコンテンツの上に、IFRAMEの要素として、透過指定して「見えなく」したコンテンツを配置することにより、ユーザーをだまして意図しない、不正な操作が可能になるという問題。「購入を取りやめる」というボタンの上に透明なボタンを配置しておけば、ユーザーは「いいえ」と選択しているつもりなのに、実際には何らかの商品を購入したことになる。
5. A Different Opera
6. Abusing HTML 5 Structured Client-side Storage
7. Cross-domain leaks of site logins via Authenticated CSS
8. Tunneling TCP over HTTP over SQL-Injection
9. ActiveX Repurposing
10. Flash Parameter Injection