CMPSCI 691ST Seminar **Fall 2011**

Lecture 13

Lecturer: Emery Berger *Scribe: Abhishek Roy*

13.1 Reflections on Trusting Trust

13.1.1 Introduction

This paper is Ken Thompson's Turing Award Lecture. Thompson got the award for his work on UNIX but in this talk describes a backdoor attack he devised. Thompson embedded the backdoor such that `login` command after compilation, would accept some password known to Thompson. His compiler could also detect when a compiler is being compiled and can insert the infected source in the new compiler's machine code. This attack is hard to detect as the source code is clean and people check the source code, not the machine code. It is not known if the infected compiler's binary was released in the wild or not.

13.1.2 Trusted Computing Base

It is hard to verify the security and correctness properties of millions of code lines in Windows and Linux. A small size of the code, is determined to be critical to the security of the system and is referred as Trusted Computing Base (TCB). Manual checking and automated theorem proving are used to examine the code of TCB. Currently, the properties of some thousand lines of code can be checked.

Microsoft has developed a secure kernel, Singularity (now called Midori), whose properties have been verified. Singularity uses message passing managed by the OS instead of shared memory. To reduce the overhead of full-copy when exchanging data, Singularity passes pointers to the data and changes the ownership of the data. *(Linux had a HTTP server compiled into the kernel to reduce copying within the kernel.)* It is easier to manage security if the underlying programming languages support properties like type safety and checks for buffer overflow etc.

13.1.3 Rootkit

Rootkit is a piece of code which has unauthorized access to system and might be invisible to OS/anti-virus software. It can be installed between the levels of Operating System and hardware. It can log keyboard signals, redirect webpages and can even run the target OS on Virtual Machine. The last type of rootkit can be detected by checking the time taken to execute CPU instructions. SubVirt is an example of virtual machine based rootkit. Rootkit can be inserted in the bootloaders also.

13.1.4 Morris Worm

Worm is a malware program which can propagate itself to other computers on the network. Morris worm, the first internet worm, was written by Robert Morris. Multiple instances of worm could run on a machine and even if a computer was infected there was a chance that it would still transfer itself to that machine,

which led to increase in network traffic. However, it only affected machines of a particular architecture. *(Tip: Don't travel with Robert Morris, he is a convicted felon and can cause delay in background checking at borders.)*

13.1.5 Attitude Towards Security and Privacy

People working in security generally argue for perfect and impractical approaches (defensive driving analogy). However, in real world people don't understand what is at risk. E.g. Paypal is linked to credit cards, bank accounts and can move money, Mint.com has access to bank accounts, what if some rogue elements buy the company. Brian Levine compares the difference with the analogy between Fort Knox (overkill) and bars on windows (can be removed through saws, but will discourage, depends on the economic gain expected).

13.1.6 Trusting Trust

Thompson says not to trust any code, you didn't write yourself. But it is very difficult to implement in practice since code is written by many organizations and hardware production is outsourced. You can use independent compilers/hardware but what if they have the same bug. For memory access, we can intercept and scramble the operations. The threat model in Thompson's paper is too powerful. At some point you have to trust someone. Unsolvable problem.

13.2 A Note on the Confinement Problem

13.2.1 Introduction

The key idea in the paper was the use of covert channels to transmit information.

13.2.2 Taint Analysis

Perl was started to replace shell scripts (it can invoke system commands) but quickly became the duct tape of the internet. A user can append e.g., `;rm -rf /` in an input string. Similarly SQL statements can be executed with conditions OR'ed with `TRUE` or a statement followed by `;DROP TABLES`. In Perl taint mode, any variable assigned to data coming from wire is marked as tainted. The taint bit is transitive and is propagated among the different program variables as the execution proceeds. The programmer is then expected to apply cleaning operations (e.g. checking for backquotes for shell commands with user input string, `;` on SQL statements) on tainted variables before sending them for execution. The tracking control/data flow should not report false positives, e.g., `( x == 1) ? z = 0 : z = 0`, if `x` is a tainted variable then, in this example, `z` is not a tainted variable.

13.2.3 Covert Channel Attacks

Timings attacks, by measuring decryption times, have been used to obtain private RSA key. Arrival time of key presses can reduce the search space for guessing passwords. If two VMs are located in the same machine then one VM, by looking at page usage, CPU usage etc., can analyze the task being done in other VM. Random delay or nonce are used to counter these attacks, but they reduce the throughput. The choice of good seed is important for random number generators. Linux has `/dev/random` file which can be used to

generate random numbers. Intel is putting a true random number generator on its chip in which an unstable circuit will provide raw bits.

13.2.4 Sandboxing

Sandboxing is a form of Software Based Fault Isolation in which an untrusted programs can run in a confined space. A sandboxed code is checked at every read/write system call. Google Native Client can run x86 code in a sandboxed browser environment. ActiveX is an early example of providing sandbox in browser.