Who Do You Think I Am?

Detecting Sophisticated Computer Attacks via USB

Lara Letaw & Kevin Butler

University of Oregon OSIRIS Lab

{zephron,butler}@cs.uoregon.edu

UNIVERSITY OF OREGON

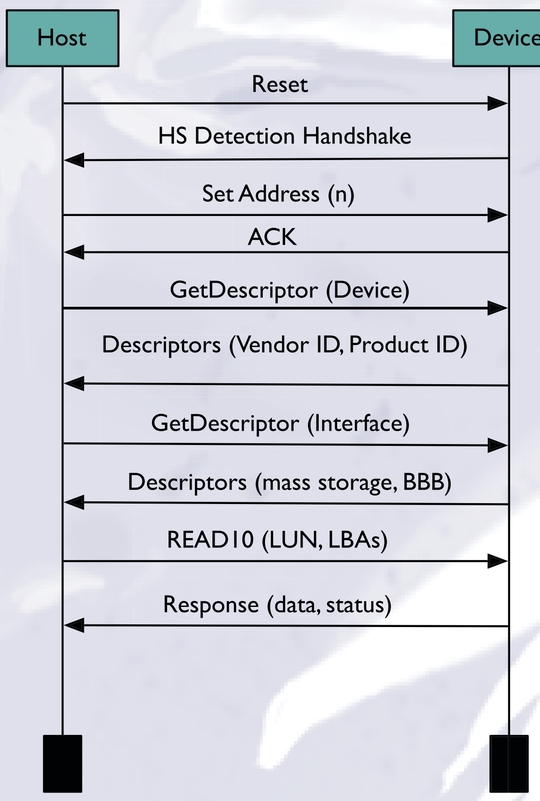
Attempts at improving computer security are often software-based, such as anti-virus and firewall programs. Since these services assume integrity of the operating system, some attacks are beyond the scope of the software's protection mechanisms. Examples of these types of attacks include:

Virtual-machine-based rootkits (VMBRs): Rootkits that avoid detection by isolating the host operating system (and all software) in a virtual machine.

Screen-spoofing: The computer interface is replaced with an interface that is identical or nearly-identical in appearance and operation, but which is actually under the adversary's control.

How can we detect that attacks of this type have occurred?

We hypothesize that analysis of USB traffic can, in some cases, reveal a compromised system.



Why is this hypothesis reasonable? Each computer has a set of hardware, firmware, and software for communicating with USB devices (the USB stack). Each component is complex, and the USB stack differs across computer types. Therefore, if any of these components are replaced, relocated, or modified, we would expect to see differences in the nature and timing of USB transactions.

Left: USB flow diagram for a mass storage device. We look at the time intervals between high-level enumeration steps (shown in the diagram), and also fine-grained timing data within each step (such as time between PING retries).

Experiment

1. Using a USB analyzer and a collection of USB devices, we first determine the expected USB traffic between a computer and each device. We sample data from 30 different machines of 12 different *types*. Machines are said to be of the same type if they are running the same operating system and have the same model identifier. We also try multiple device types: human interface (USB 1.1 mouse), mass storage (USB 2.0 thumb drive), and video (USB 2.0 webcam). Our final data set contains 1102 4-dimensional points of timing data. After observing that use of the thumb drive resulted in the largest and most stable set of USB transactions, the thumb drive became our sole test device. Thus, our data set total does not include the webcam or mouse.

The USB analyzer monitors the connection between the device and the test computer. The observed transactions are sent to analysis software on a second computer.

2. We then change a component of the computer (this is how we simulate an adversary) and see if the USB traffic changes. We use VMWare to launch virtual Windows (within Windows) and Ubuntu (within Ubuntu) on some of the computers, and again examine the USB traffic. This corresponds to the VMBR attack. We also examine the effects of launching the operating system from a Live CD. This corresponds to the screen-spoofing attack.

Results

1. USB traffic is consistent between computers of the same type. This means we have an expected USB fingerprint for each computer type we examined.

Learner	Accuracy	IS	Brier	AUC
Naïve Bayes	0.902	3.097	0.140	0.989
Decision Tree	0.978	3.354	0.042	0.999
KNN(10)	0.955	3.310	0.070	0.998
KNN(9)	0.955	3.308	0.070	0.998
KNN(8)	0.961	3.304	0.071	0.998
KNN(7)	0.955	3.298	0.072	0.998

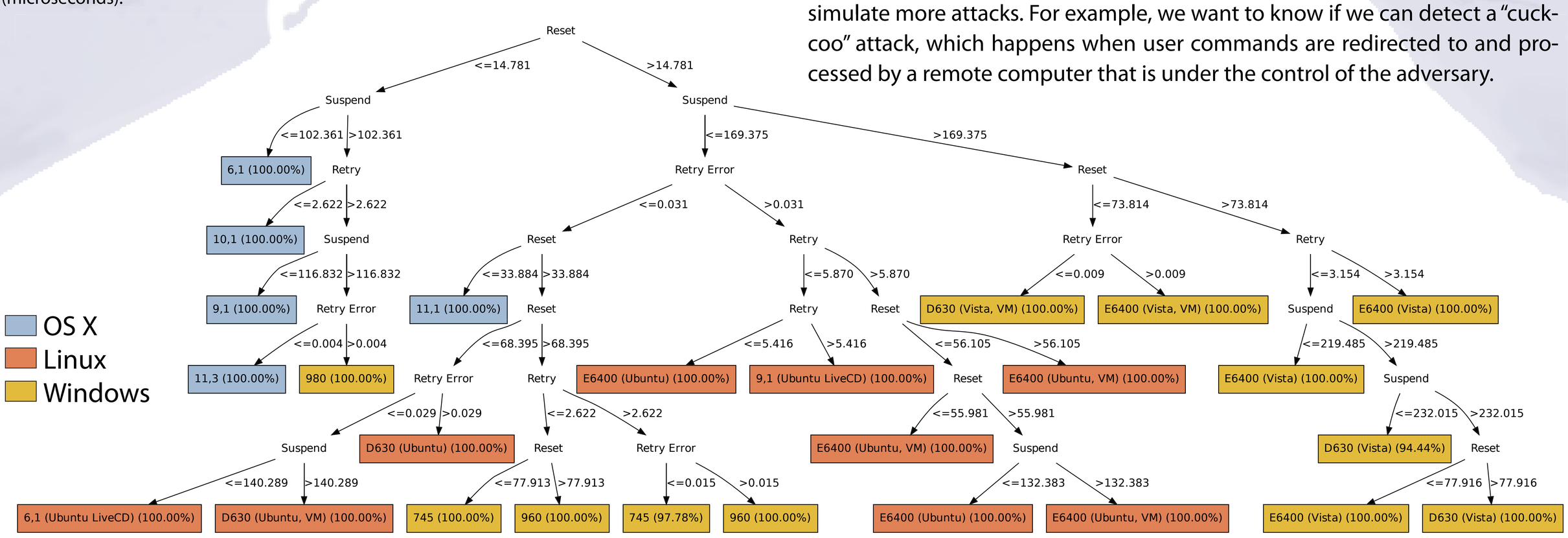
Above: Cross-validation of three classification techniques: Naive Bayes, decision trees, and k-nearest-neighbors (with k=7, 8, 9, and 10). High accuracy, information score (IS), and area under ROC curve (AUC), and low Brier Score indicate that the technique is successful at classifying our data.

2. USB traffic differs between computers of different types, and is also different for machines running a virtual machine or Live CD. This means we may be able to detect that a sophisticated attack has taken place.

Future Work

Our work is just beginning. We want to look at many other computer types, and simulate more attacks. For example, we want to know if we can detect a "cuckoo" attack, which happens when user commands are redirected to and processed by a remote computer that is under the control of the adversary.

Below: Decision tree showing the classification of 12 computer types and 6 representations of compromised systems. Four USB timing attributes were identified and used to distinguish between computers: duration of Suspend state (milliseconds), duration of Reset state (milliseconds), time between IN transaction retries (microseconds), and variance in the time between retries (microseconds).



Background artwork courtesy of Sèanar Letaw.