DIDS: Integrated host and network monitoring, live taps, lateral tracking, oh... and all in 1991

Todd Heberlein
LTH@NetSQ.com
20 Sep 2012

From 1990 to 1992 UC Davis, Haystack Labs, and Lawrence Livermore National Laboratory (LLNL) worked on the Distributed Intrusion Detection System (DIDS). We handed it off to the United States Air Force (USAF), which funded the work. The USAF planned to roll it out across the entire Air Force network. They hired Trident Data Systems (TDS) to "productize" our prototype and support the roll out. Eventually the network monitoring portion (my NSM) and a centralized Director was rebranded ASIM and rolled out Air Force wide. As far as I know, the ASIM sensor grid was the first global-scale intrusion detection system. This document shows some screenshots of DIDS at about the midpoint of its development.

1 Introduction

By 1990 I was starting to get good results from my Network Security Monitor (NSM) at UC Davis. Meanwhile Haystack Labs was shipping their Haystack audit trail intrusion detection system to the Air Force. UC Davis and Haystack Labs teamed up, with Lawrence Livermore National Laboratory (LLNL) as the prime contractor, to develop the Distributed Intrusion Detection System (DIDS) — an intrusion detection system that combined the analyses of both audit-based and network-based intrusion detection sensors.

DIDS combined multiple sensor types (network and host audit trails), signature and anomaly detection, and distributed analysis (analysis was performed by host monitors, the network monitor, and a centralized expert system). DIDS could aggregate subtle behavior by a user as he moved laterally within the network, so while on any single host or network connection his activity might remain below the radar, aggregating all the low-level warning messages generated by each sensor could identify the suspicious behavior. The analyst at the DIDS console could tap into live network connections or audit data to see exactly what the person was doing in real time.

A few years ago I ran across some old 35mm slides I took of the DIDS Director's monitor during a demo. Some lawyers helpfully scanned them in (actually they became evidence in a patent lawsuit), but now that that affair is in the past, I've decided to post the pictures here.

2 Main Window: NIDs, Hosts, and Debugging

Figure 1 shows DIDS' main window. The top-left tracked all the Network IDs (NIDs), displaying each NID number and a horizontal bar graph showing its level of suspiciousness. At this point, six users were logged in. Nothing suspicious is going on yet, so they all have short green bars.

On the right side are icons for each host with a host monitor. The main area is the debugging information describing what the expert system is thinking. (Hey, this was a prototype, so this debugging information was important to us)

The Network ID (NID) represented a user logged into the network. When the user first logged in, either at the console or via a remote login (at the time, that meant telnet or rlogin), he was assigned a NID. That NID would track that user as he moved through the network. If the user SUed to a different user or logged into a different machine (under either the same or different user name), all the activity was mapped back to the same NID. In other words, the NID was the mechanism for aggregating all the user's activity across the network.

The expert system running on the DIDS Director was responsible for mapping activity reported by each individual monitor (both network and host monitors) to the same NID. Then future messages of even the slightest suspiciousness reported by any of the sensors (e.g., a string in a network connection picked up by the network monitor or anomalous browsing behavior picked up by the host monitor) would be aggregated to the NID.

Oh, and checkout the timestamp information. We were conducting this demo on Sep 20, 1991.

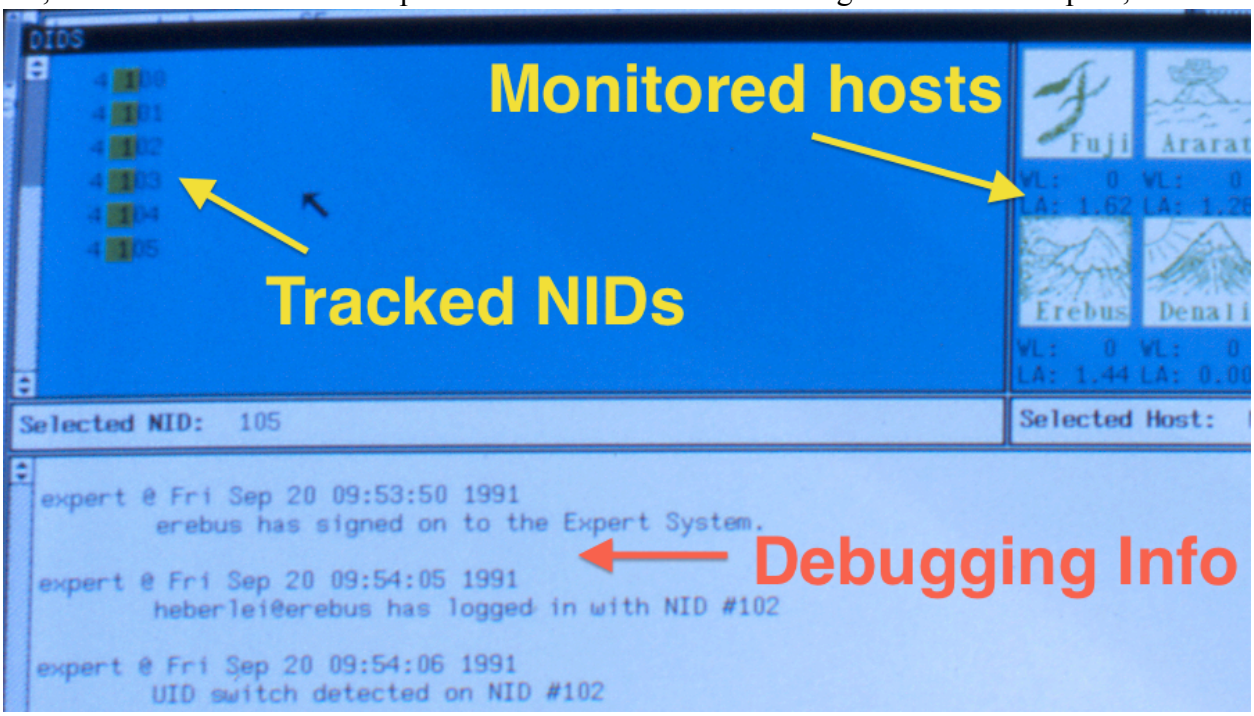

Figure 1: Main window

3 Aggregated Suspicious Behavior – Browsing

Figure 2 shows that NID 106 has increased its suspiciousness level to 40, which causes the horizontal bar to grow and change color to red. An alert about a “suspicious session” is sent to the screen; the user appears to be “Browsing”. In other words, the user has looked at a lot more

files than usual. The browsing behavior is aggregated from multiple hosts, so the behavior on any individual host may remain below the radar.

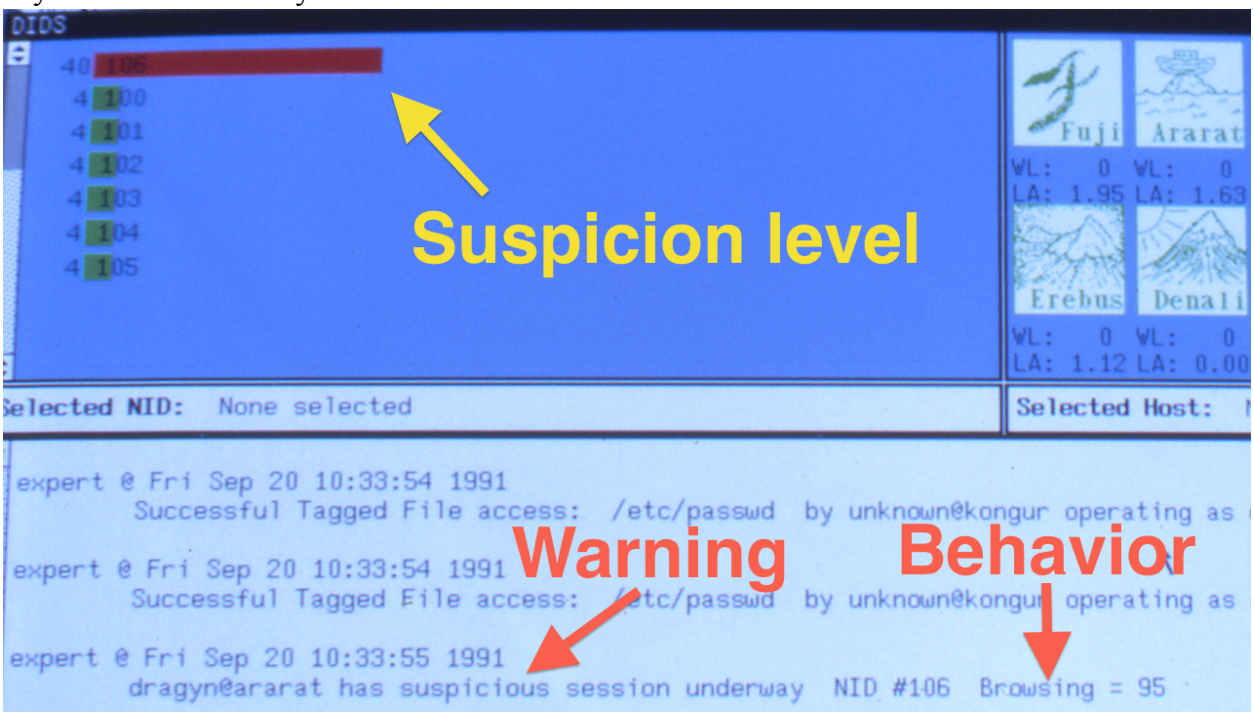

Figure 2: General suspicious behavior – Browsing

4 Penetration and Lateral Movement

Figure 3 shows a user's movement as they move through the network. The user initially entered the monitored domain from an outside host at the bottom (labeled “others”). The user logged into ararat and then moved laterally to hosts fuji and erebus.

The user has selected the last connection in this path, the one between the hosts fuji and erebus. The summary information at the bottom says the user names used on both hosts were "dragyn". On the right is a button to let the user “Wiretap” the connection.

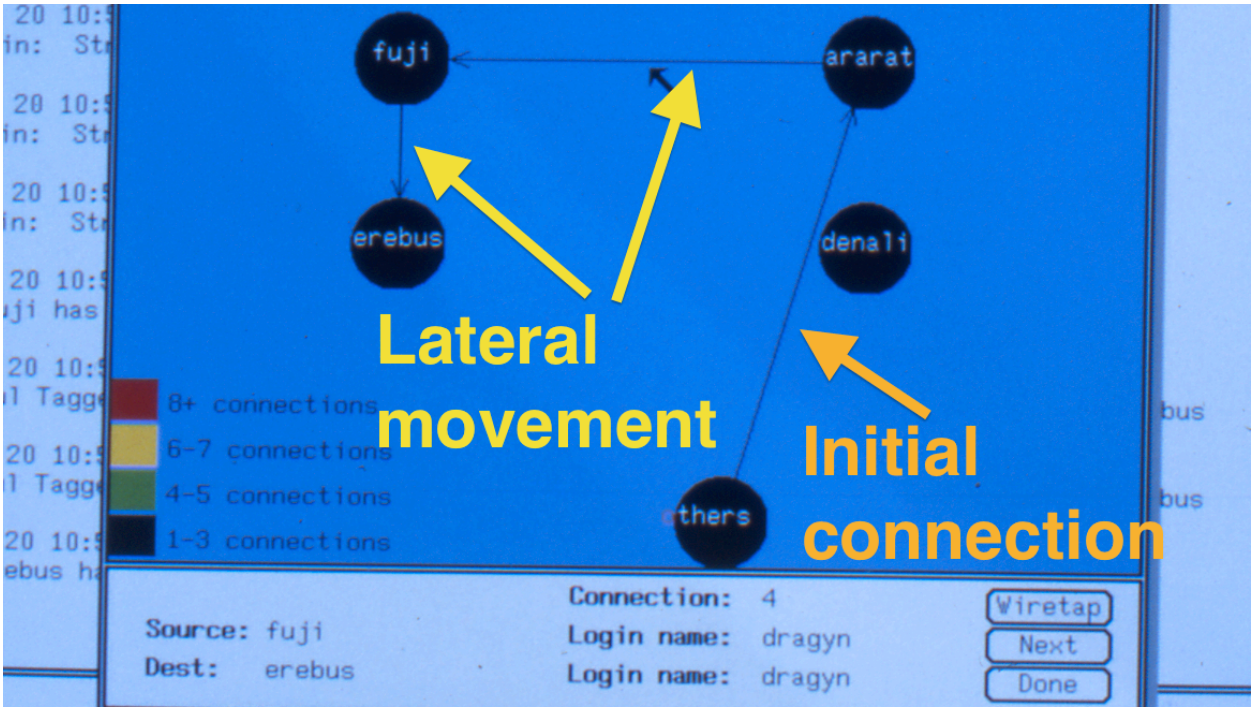

Figure 3: Tracking lateral movement after penetration

5 Tapping a Login Connection

Figure 4 shows the wiretap window. We could tap the input direction (what the user typed) or the output direction (what the user saw on his screen). Here we have selected the output information on connection 2. The user entered the command “whoami”, and the output was “dragyn”. The user then enters the “ls” command to list the current directory's content. There is only one file, “hello”, in the directory.

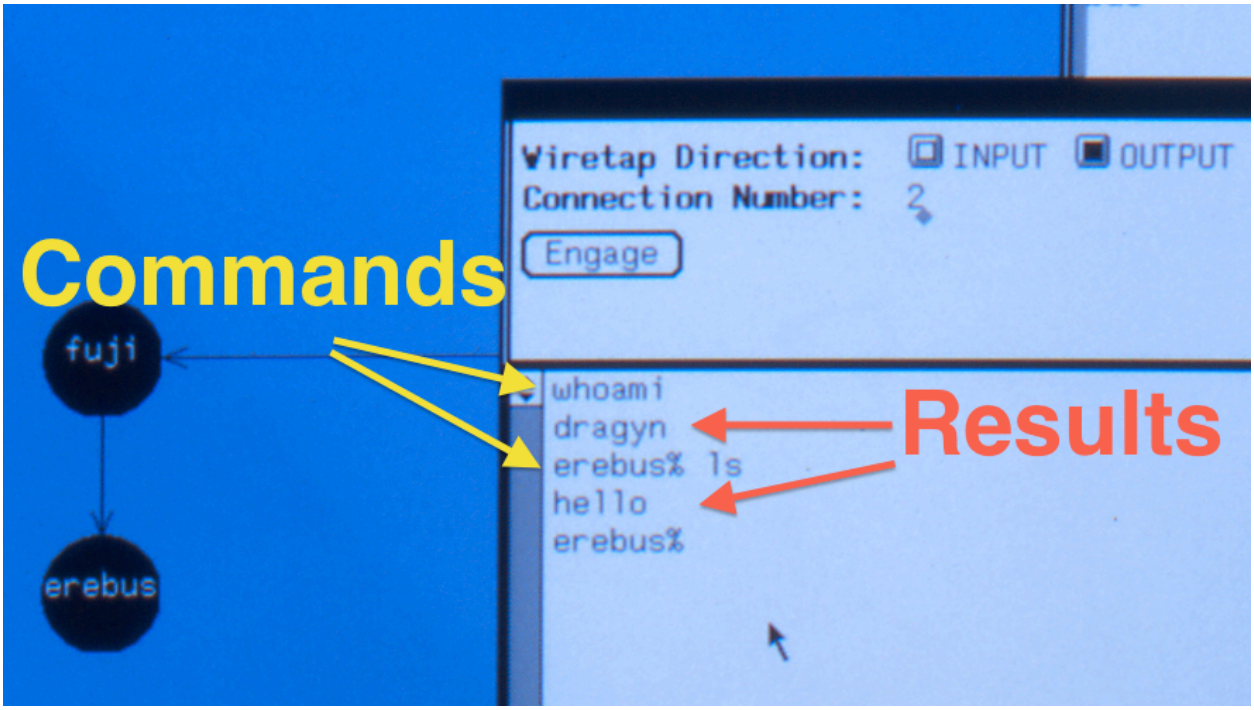

Figure 4: Network taps

6 Audit Taps

In addition to network taps, we could do audit taps where audit records associated with a NID are displayed. Figure 5 shows both the audit and network taps at the same time. In this figure I've highlighted the program execution reported in the audit trail and the corresponding command shown in the network connection.

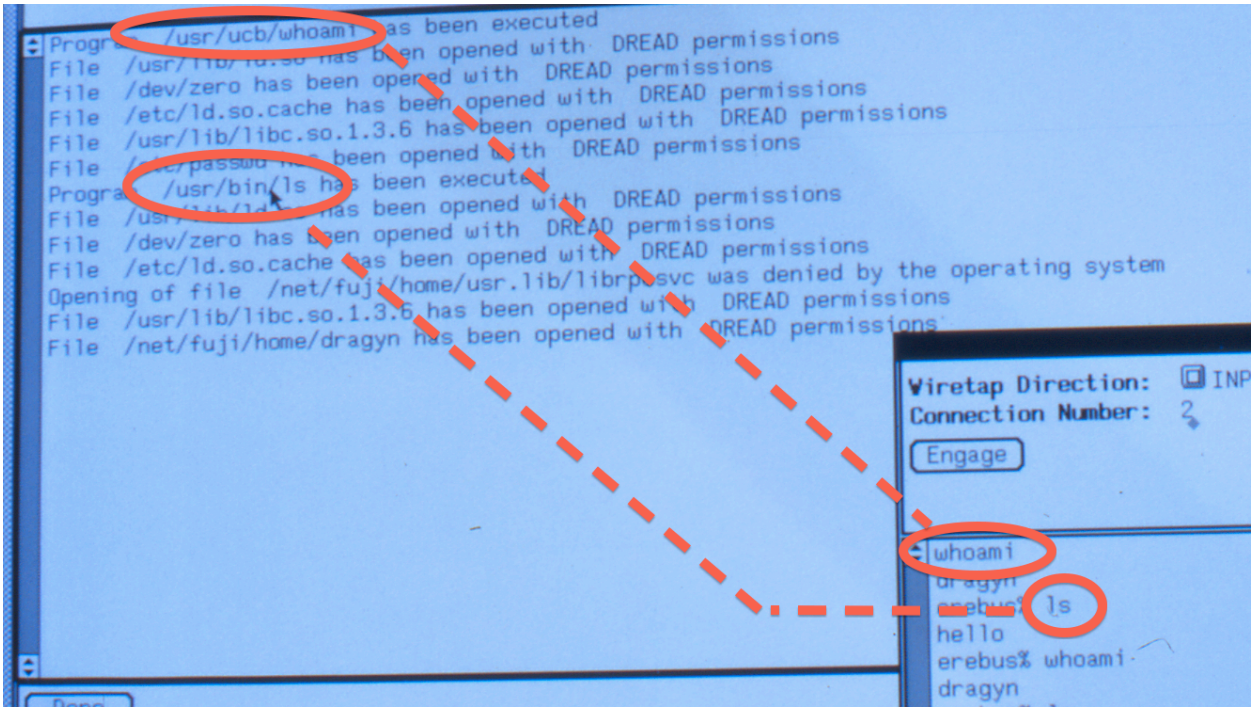

Figure 5: Both audit and network taps

7 Conclusions

I'd forgotten how advanced DIDS was at the time. This was a time when, I believe, DOS was still the most used operating system. When we designed DIDS we had assumed C2 level audit trails would be running on all the government computers by the end of the project in 1992. After all, the "Computer Security Act of 1987" said that they should be within five years, hence the rallying cry of "C2 by 92". Lesson learned: don't believe that a legislative mandate will actually cause things to change.