Host-based intrusion detection system

A **host-based intrusion detection system** (**HIDS**) is an intrusion detection systemthat is capable of monitoring and analyzing the internals of a computing system as well as the network packets on its network interfaces, similar to the way a network-based intrusion detection system (NIDS) operates. This was the first type of intrusion detection software to have been designed, with the original target system being the mainframe computer where outside interaction was infrequent.

Overview

A host-based IDS is capable of monitoring all or parts of the dynamic behavior and the state of a computer system, based on how it is configured. Besides such activities as dynamically inspecting network packets targeted at this specific host (optional component with most software solutions commercially available), a HIDS might detect which program accesses what resources and discover that, for example, a word-processor has suddenly and inexplicably started modifying the system password database. Similarly a HIDS might look at the state of a system, its stored information, whether in RAM, in the file system, log files or elsewhere; and check that the contents of these appear as expected, e.g. have not been changed by intruders.

One can think of a HIDS as an agent that monitors whether anything or anyone, whether internal or external, has circumvented the system's security policy.

Monitoring dynamic behavior

Many computer users have encountered tools that monitor dynamic system behaviour in the form of anti-virus (AV) packages. While AV programs often also monitor system state, they do spend a lot of their time looking at who is doing what inside a computer – and whether a given program should or should not have access to particular system resources. The lines become blurred here, as many of the tools overlap in functionality.

Some intrusion prevention systems protect against buffer overflow attacks on system memory and can enforce security policy.

Monitoring state

The principle operation of a HIDS depends on the fact that successful intruders (hackers) will generally leave a trace of their activities. In fact, such intruders often want to *own* the computer they have attacked, and will establish their "ownership" by installing software that will grant the intruders future access to carry out whatever activity (keystroke logging, identity theft, spamming, botnet activity, spyware-usage etc.) they envisage.

In theory, a computer user has the ability to detect any such modifications, and the HIDS attempts to do just that and reports its findings.

Ideally a HIDS works in conjunction with a NIDS, such that a HIDS finds anything that slips past the NIDS. Commercially available software solutions often do correlate the findings from NIDS and HIDS in order to find out about whether a network intruder has been successful or not at the targeted host.

Most successful intruders, on entering a target machine, immediately apply best-practice security techniques to secure the system which they have infiltrated, leaving only their own backdoor open, so that other intruders can not take over *their* computers.

Technique

In general a HIDS uses a database (object-database) of system objects it should monitor – usually (but not necessarily) file system objects. A HIDS could also check that appropriate regions of memory have not been modified – for example, the system call table for Linux, and various vtable structures in Microsoft Windows.

For each object in question a HIDS will usually remember its attributes (permissions, size, modifications dates) and create a checksum of some kind (an MD5, SHA1 hash or similar) for the contents, if any. This information gets stored in a secure database for later comparison (checksum database).

An alternate method to HIDS would be to provide NIDS type functionality at the network interface (NIC) level of an end-point (either server, workstation or other end device). Providing HIDS at the network layer has the advantage of providing more detailed logging of the source (IP address) of the attack and attack details, such as packet data, neither of which a dynamic behavioral monitoring approach could see.

Operation

At installation time – and whenever any of the monitored objects change legitimately – a HIDS must initialize its checksum-database by scanning the relevant objects. Persons in charge of computer security need to control this process tightly in order to prevent intruders making un-authorized changes to the database(s). Such initialization thus generally takes a long time and involves cryptographically locking each monitored object and the checksum databases or worse. Because of this, manufacturers of HIDS usually construct the object-database in such a way that makes frequent updates to the checksum database unnecessary.

Computer systems generally have many dynamic (frequently changing) objects which intruders want to modify – and which a HIDS thus should monitor – but their dynamic nature makes them unsuitable for the checksum technique. To overcome this problem, HIDS employ various other detection techniques: monitoring changing file-attributes, log-files that decreased in size since last checked, and numerous other means to detect unusual events.

Once a system administrator has constructed a suitable object-database – ideally with help and advice from the HIDS installation tools – and initialized the checksum-database, the HIDS has all it requires to scan the monitored objects regularly and to report on anything that may appear to have gone wrong. Reports can take the form of logs, e-mails or similar.

Protecting the HIDS

A HIDS will usually go to great lengths to prevent the object-database, checksum-database and its reports from any form of tampering. After all, if intruders succeed in modifying any of the objects the HIDS monitors, nothing can stop such intruders from modifying the HIDS itself – unless security administrators take appropriate precautions.

Many worms and viruseswill try to disable anti-virus tools, for example.

Apart from crypto-techniques, HIDS might allow administrators to store the databases on a CD-ROM or on other read-only memory devices (another factor in favor of infrequent updates...) or storing them in some off-system memory. Similarly, a HIDS will often send its logs off-system immediately – typically using VPN channels to some central management system.

One could argue that the trusted platform module comprises a type of HIDS. Although its scope differs in many ways from that of a HIDS, fundamentally it provides a means to identify whether anything/anyone has tampered with a portion of a computer. Architecturally this provides the ultimate (at least at this point in time) host-based intrusion detection, as depends on hardware external to the CPU itself, thus making it that much harder for an intruder to corrupt its object and checksum databases.

Reception

InfoWorld states that host-based intrusion-detection system software is a useful way for network managers to find malware, and suggest they run it on every server, not just critical servers.