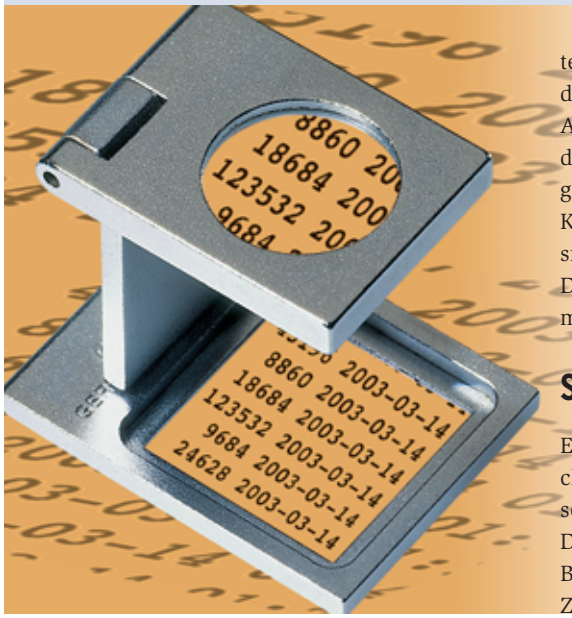
AIDE: Host-basiertes IDS im Tripwire-Stil

Hilfe bei der Spurensuche

Eindringlinge an ihren Spuren erkennen – das ist das Ziel des Advanced Intrusion Detection Environment. In einer Datenbank hinterlegt AIDE die Attribute und kryptographischen Prüfsummen wichtiger Files und Verzeichnisse. So bemerkt es jede nicht autorisierte Änderung und schlägt Alarm. Thorsten Scherf



Wie wichtig und erfolgreich ein Intrusion-Detection-System sein kann, erwies sich Ende 2003 nach dem Einbruch in mehrere zentrale Debian-Server [4]. Der Eindringling hatte eine bis dahin unbekannte Sicherheitslücke im Linux-Kernel genutzt und auf den geknackten Maschinen ein Rootkit installiert. Danach liefen die Server nicht mehr stabil. Dass für die Instabilität nicht irgendein technisches Problem, sondern ein Einbruch verantwortlich war, darauf deuteten die Warnungen von AIDE hin: Das Advanced Intrusion Detection Environment [1] bemerkte, dass »/sbin/init« ersetzt worden war und sich die »mtime«- und »ctime«-Werte für »/usr/lib/locale/en_US« geändert hatten.

Solche Spuren sind typisch für unerwünschte Besucher. Host-basierte Intrusion Detection Systems (HIDS) suchen diese Hinterlassenschaften und reagieren darauf. Im Unterschied dazu belauscht ein NIDS (Netzwerk-IDS) die Datenübertragung im LAN und versucht darin Angriffsspuren zu erkennen.

AIDE gehört zur Klasse der HIDS und dort zur Untergruppe SIV (System Integrity Verifier). Vom Vorreiter dieser Klasse – Tripwire [3] – unterscheidet es sich unter anderem durch seine Größe. Das Leichtgewicht AIDE umfasst gerade mal 800 KByte.

Soll-Zustand speichern

Eine Konfigurationsdatei bestimmt, welche Datei- und Ordner-Attribute AIDE in seiner Datenbank speichern soll. Diese Datenbank legt den Soll-Zustand fest. Beispiele für Datei-Attribute sind Inodes, Zugriffsrechte, die Größe oder die Anzahl der Hardlinks. Außerdem berechnet das System kryptographische Checksummen der zu überprüfenden Objekte und speichert diese ebenfalls. Die Soll-Datenbank muss einen sicheren Zustand beschreiben. Das gelingt am ehesten bei einer frischen Betriebssysteminstallation, deren Files aus einer vertrauenswürdigen Quelle stammen.

Der Admin kann in der Folgezeit AIDE beliebig häufig aufrufen und den Ist-Zustand mit der Soll-Datenbank vergleichen lassen. Nach einem Einbruch installieren Angreifer häufig so genannte Rootkits [5], die ihnen uneingeschränkten Zugang zum System geben und sicherstellen, dass sie jederzeit erneut einbrechen können. Dazu ersetzen oder manipulieren Rootkits vorhandene Dateien. Ohne IDS fällt dieser Eingriff kaum auf, da die Ersatzprogramme äußerlich dem Original meist exakt gleichen. Ihre Zusatzfunktion ist gut versteckt und verhindert, dass Admins den Eindringling bemerken.

Ein Integritätsprüfer schafft Abhilfe, indem er die Manipulation aufdeckt. Um alle zur Verfügung stehenden kryptographischen Checksummen einsetzen zu können, ist neben den AIDE-Sourcen [1] zusätzlich das Mhash-Paket [2] nötig. Zur Installation genügt der bekannte Dreisatz »./configure && make && make install«. Die Konfigurationsdatei (siehe Listing 1) lautet »/etc/aide.conf« und ist direkt nach der Installation an das eigene System anzupassen.

Diese Beispielkonfiguration beginnt mit einigen Parametern (Zeilen 1 bis 4), die unter anderem festlegen, wo AIDE seine Datenbank ablegt. Die Zeilen 5 bis 7 definieren neue Gruppen von zu überprüfenden Attributen (siehe Tabelle 1). Wegen dieser Ketten fallen die Prüfregeln recht übersichtlich aus.

Regeln für die Prüfung

Bei den Prüfregeln ist das Objekt anzugeben, gefolgt von den Attributen, die AIDE überwachen soll (Zeile 18). Dabei ist größte Sorgfalt gefragt. Sind zu wenig Objekte oder unzureichende Attribute

AIDE

Name: AIDE, Advanced Intrusion Detection Environment
Lizenz: GPL
Status: Beta (Version 0.10)
Systeme: Linux, BSD, OpenBSD, FreeBSD, Solaris, Unixware, AIX, TRU64, Cygwin
Kategorie: Gehört zur Klasse der Host-basierten Intrusion-Detection-Systeme (IDS), genauer: System Integrity Veryfier (SIV)
Aufgaben: Überprüft die Integrität der Systemdateien und stellt Modifikationen an den Konfigurationsfiles fest
Homepage: [http://www.cs.tut.fi/~rammer/aide.html]

Tabelle 1: AIDE-Attribute

Attribut	Bedeutung
p	Permissions (Zugriffsrechte)
i	Inode
n	Number of links (Anzahl der Hardlinks)
u	User
g	Group
s	Size (Größe)
m	Mtime (Modifikation des Datei-Inhalts)
a	Atime (Access, Zugriff)
c	Ctime (Change, Änderung der Inode-Information)
S	Growing Size (wachsende Größe)
md5	MD5-Checksumme
sha1	SHA1-Checksumme
rmd160	RMD160-Checksumme
tiger	Tiger-Checksumme
R	p+i+n+u+g+s+m+c+md5
L	p+i+n+u+g
E	Empty group
>	Wachsendes Logfile (p+u+g+i+n+S)
Nur bei installiertem Mhash	
crc32	CRC32-Checksumme
haval	Haval-Checksumme
gost	Gost-Checksumme

angegeben, fällt ein Einbruch unter Umständen nicht auf. Es ist auch darauf zu achten, dass keine Objekte überprüft werden, die auf dem eigenen System gar nicht existieren. AIDE würde direkt das Fehlen von Objekten in der Datenbank vermerken. Eine Übersicht der möglichen Attribute zeigt Tabelle 1.

Das Kommando »aide --init« erzeugt die Soll-Datenbank und schreibt sie in das Verzeichnis »/var/lib/aide«. Es ist ratsam, die Datenbank sowie die Konfigurationsdatei an einen sicheren Ort zu kopieren, beispielsweise auf eine CD. Ein Angreifer darf diese Files nicht manipulieren können.

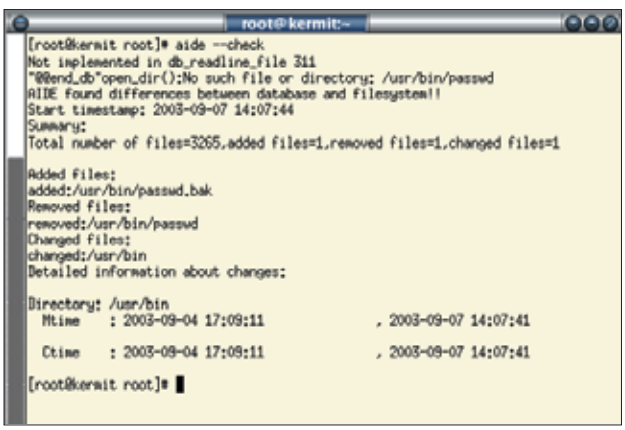


Abbildung 1: AIDE stellt hier fest, dass die Datei »/usr/bin/passwd« gelöscht wurde. Der Admin muss nun entscheiden, wie er auf diese Manipulation reagiert.

Vergleiche anstellen

Ist die Konfiguration erfolgreich abgeschlossen, beginnt »aide --check« mit dem Überprüfen des Systems. Es bietet sich an, dieses Kommando regelmäßig per Cron zu starten. AIDE liest die Datenbank und vergleicht sie mit dem Ist-Zustand der Dateien und Verzeichnisse. Meldet es Änderungen an den Objekten (Abbildung 1), muss der Admin entscheiden, ob es sich um eine gewollte Änderung oder um einen nicht autorisierten Zugriff handelt. Im ersten Fall ist die Datenbank anzupassen, um künftig keinen Fehlalarm mehr auszulösen. Dazu ruft er »aide --update« auf.

Sind die Änderungen nicht gewollt, muss er zunächst überprüfen, um welche Regelverletzung es sich handelt und wieso diese möglich war. Wurden Dateien ohne Wissen des Systemverwalters ausgetauscht oder modifiziert, deutet vieles auf ein Rootkit oder ein Trojanisches Pferd hin. Er sollte nun geeignete Gegenmaßnahmen einleiten [6], [7].

AIDE und Tripwire

Beim Vergleich von AIDE mit dem bekannten und deutlich älteren Tripwire fällt der Größenunterschied auf. AIDE ist mit rund 800 KByte ein echtes Leichtgewicht. So ist es ohne weiteres möglich, eine komplette Installation inklusive Datenbank auf einer Diskette zu platzieren. Die Konfiguration geht sehr schnell von der Hand.

Allerdings reichen die Berichtsfunktionen nicht an Tripwire heran. So ist es nicht möglich, sich einen Report per E-Mail zusenden zu lassen. Nach einer Möglichkeit, die Datenbank und die Konfigurationsdatei zu verschlüsseln, sucht man ebenfalls vergeblich. Ein Blick auf die To-do-Liste lässt aber viele neue Funktionen erwarten, zumal das Projekt seit kurzem wieder mehr Fahrt aufnimmt. Aktuelle Informationen finden sich auf der Mailingliste des Projekts [8]. *(fjl)*

Infos

[1] AIDE-Homepage: [http://www.cs.tut.fi/~rammer/aide.html]
[2] Mhash: [http://mhash.sf.net]
[3] Tripwire: [http://www.tripwire.org] sowie Thorsten Scherf, „Spurensucher: Tripwire“: Linux-Magazin Security Edition 1/04, S. 60
[4] Debian-Untersuchungsbericht nach Serverkompromittierungen: [http://www.linux-community.de/story?storyid=11004]
[5] Oktay Altunergil, „Understanding Rootkits“: [http://linux.oreillynet.com/pub/a/linux/2001/12/14/rootkit.html]
[6] Keith J. Jones, „Spurensuche – Incident Response, die richtige Reaktion auf einen Einbruch“: Linux-Magazin 3/02, S. 48
[7] Ralf Spenneberg, „Intrusion Detection für Linux-Server“: Markt und Technik 2002
[8] AIDE-Mailinglist-Archiv: [http://www.mail-archive.com/aide@cs.tut.fi/]

Der Autor

Thorsten Scherf arbeitet für die GfN Training GmbH, er führt dort Red-Hat-Schulungen durch. Seine Schwerpunkte sind Open Source und Netzwerksicherheit.

Listing 1: AIDE-Konfigurationsdatei

```
01 database=file:/var/lib/aide/aide.db
02 database_out=file:/var/lib/aide/aide.db.new
03 verbose=20
04 report_url=stdout
05 All=R+a+sha1+rmd160+tiger
06 Norm=s+n+b+md5+sha1+rmd160+tiger
07 R=p+i+n+u+g+s+m+c+md5
08
09 # Folgende Verzeichnisse nicht überwachen
10
11 !/dev
12 !/tmp
13 !/proc
14 !/usr/src
15
16 # Das komplett Root-Verzeichnis kontrollieren
17
18 / R
```