Ein Netzwerk von IDS-Sensoren für Angriffsstatistiken

Till Dörges, Olaf Gellert, Klaus-Peter Kossakowski

Abstract: Angriffserkennende Systeme (Intrusion Detection Systems, IDS) sind mittlerweile eine etablierte Technik, um Informationen über die Gefährdung der eigenen Systeme zu gewinnen. Moderne IDS realisieren ein Framework einer Vielzahl unterschiedlicher Sensoren, um eine möglichst vollständige Datenerfassung durchzuführen. Der Einsatz vieler Sensoren, die über das gesamte Internet verteilt sind, ermöglicht die Beobachtung der unterschiedlichen Angriffscharakteristiken in den verschiedenen Teilnetzen. Durch Korrelation der Angriffe auf die Sensoren lassen sich auch Rückschlüsse auf das Verhalten von Angreifern und auf die Verbreitungsstrategien von Würmern und Viren ziehen. Im Rahmen des Projektes `eCSIRT.net` wurde ein Netz aus international verteilten Sensoren aufgebaut, das die gesammelten Angriffsdaten zur Erzeugung von entsprechenden Statistiken nutzt. Weiterhin können die gesammelten Daten den Netzbetreibern behilflich sein, Vorfälle zu erkennen und zu bearbeiten. Dieser Artikel stellt die Realisierung des Sensor-Netzwerkes vor und zeigt erste Ergebnisse der Datensammlung.

Schlüsselworte: Incident Handling, Angriffs-Statistiken, Intrusion Detection, Distributed IDS, Sensor-Netzwerk.

1 Einleitung

Eines der Ziele des von der EU geförderten `eCSIRT.net` Projekts (vgl. [eCa]) war die Erstellung von Statistiken, um Einblicke in die Arbeit von Computer Notfall Teams zu gewinnen und um das öffentliche Bewusstsein für die Gefahren des Internets zu schärfen. Letzteres wurde auf zweierlei Arten realisiert: Zum Einen durch manuelle Dateneingabe der teilnehmenden CERTs, z. B. Arbeitslast, Vorfallscharakteristika, zum Anderen durch den Einsatz eines verteilten Intrusion Detection Systems (IDS) realisiert, das ein automatisches Sammeln von Angriffsdaten übernimmt. Hierbei wurden in verschiedenen Netzwerken in ganz Europa jeweils sog. IDS-Sensoren aufgesetzt, die Netzwerkverkehr analysieren, Angriffe erkennen und entsprechende Warnmeldungen an einen zentralen Manager versenden (vgl. [GDK04]). Die auf diese Weise gesammelten Daten werden zum Erzeugen von Statistiken verwendet.

Im folgenden Abschnitt wird zunächst der Aufbau des Sensor-Netzwerks beschrieben. Es wird dabei sowohl auf die Planung und Konzeption als auch auf die Implementierung eingegangen. Darauf werden die bisherigen Erfahrungen mit dem Betrieb des Sensor-Netzwerks vorgestellt. Erste Auswertungsergebnisse der gesammelten Daten folgen in Abschnitt 4. Zuletzt wird eine Zusammenfassung der präsentierten Ergebnisse und ein Ausblick auf zukünftige Entwicklungen gegeben.

2 Aufbau des Sensor-Netzwerks

Zunächst mussten die organisatorischen Rahmenbedingungen geklärt werden, die für eine Datensammlung notwendig sind. Die Ergebnisse der ersten Gespräche mit allen teilnehmenden Teams wurden in einer Policy festgehalten, die den Prozess der Datensammlung verbindlich regelt. In einer zweiten Phase wurde die Realisierung des Sensor-Netzwerks angegangen. Beide Phasen sollen in den nächsten beiden Abschnitten genauer vorgestellt werden.

2.1 Planung und Konzeption

Der Aufbau des Sensor-Netzwerks erforderte zunächst die Klärung vieler organisatorischer Einzelheiten. Hierzu gehören sowohl praktische Aspekte (Ansprechpartner für jede teilnehmende Organisation, Sicherstellen von vertraulicher Kommunikation etc.) als auch rechtliche Gegebenheiten und Anforderungen bezüglich des Datenschutzes. In dieser Phase kristallisierten sich auch bereits grundlegende Vorgaben für die Realisierung des Netzwerks heraus. Ein Beispiel hierfür ist direkt mit der Art der Datensammlung verknüpft: Auch wenn alle teilnehmenden Organisationen aus dem Bereich der Nationalen Forschungsnetze stammten, die i. d. R. eine weniger zurückhaltende Informationspolitik betreiben als kommerzielle Institutionen, wurden grundlegende Bedenken gegenüber einer netzweiten Datensammlung durch die Sensoren geäußert. Daraus folgte unmittelbar für die Konzeption der Sensoren, dass nur Netzwerkverkehr analysiert werden darf, der direkt an einen Sensor gerichtet ist, und dass für die Sensoren dedizierte, nicht-operative Systeme verwendet werden. Der Sensor wird dadurch zu einer Art Honeypot, es werden keine Daten aufgezeichnet, die an andere Systeme im gleichen Netzwerksegment gerichtet sind. Hieraus ergibt sich einerseits eine Beschränkung der Fähigkeiten der Sensoren, da nun Angriffe auf Systeme im normalen Netzwerkbetrieb nicht beobachtet und somit nicht erkannt werden können. Andererseits ermöglicht diese Vorgehensweise erst den Einsatz von Sensoren innerhalb von fremden Produktionsnetzen, da sonst Bedenken bezüglich der Privatsphäre von Benutzern und des Schutzes von Unternehmensdaten überwiegen könnten. Ebenso erleichtert diese Konzeption das Aufsetzen von Sensoren, da keine weiteren Änderungen der Netzwerkumgebung vorgenommen werden müssen, sonst wäre beispielsweise das Konfigurieren eines SPAN-Ports am zugehörigen Switch notwendig, um dem Sensor zu ermöglichen, in einem geswitchten Netzwerk auch den Verkehr anderer Systeme zu sehen. Die vorigen Überlegungen machen deutlich, dass eine Policy, die sowohl den Prozess der Datensammlung als auch die organisatorischen Maßnahmen eindeutig regelt, notwendig ist.

In der Policy wurden die folgenden Vorgaben und Regelungen für das Sensor-Netzwerk festgehalten:

Voraussetzungen für die Teilnahme:

- Die Teilnahme an der Datensammlung ist beschränkt auf Notfallteams, zu de-

nen bereits ein Vetrauenspfad besteht (entweder durch die Teilnahme am Projekt `eCSIRT.net` oder aber durch die Akkreditierung des Trusted Introducer Dienstes (siehe [KS00] und `http://www.ti.terena.nl/`). Somit existiert auch ein vertrauenswürdiger Kommunikationspfad, so dass die Zugangsdaten für die Sensoren sicher verteilt werden können.

- Weitere Kontaktinformationen sind notwendig für den Betreuer des Sensors. Eine weitere Voraussetzung ist die kryptographische Absicherung von übermittelten Daten, sowohl bezüglich der Authentizität als auch der Vertraulichkeit.

Austausch der gesammelten Informationen:

- Beschreibung der gesammelten Informationen. Wesentliches Kriterium ist hier die Beschränkung auf an den Sensor gerichtete Daten.
- Beschreibung des Informations-Austausches zwischen den Sensoren und dem zentralen Sammelpunkt. Authentisierung und Verschlüsselung sind notwendig, um die vertraulichen Daten vor den Augen Dritter zu schützen und die Integrität der Daten sicherzustellen.
- Einschränkung der Veröffentlichung der gesammelten Daten. Eine Veröffentlichung der internen Daten erfordert die Zustimmung des Teams, von dem die Daten stammen. Für die Öffentlichkeit bestimmte Daten und Auswertungen werden im Vorwege festgelegt.

Durch die Abstützung auf den Trusted Introducer Dienst wurde eine einfache Übermittlung von vertraulichen Informationen ermöglicht. Dies ist notwendig, um die für die Absicherung der Datensammlung benötigten Zertifikate und Secret Keys an die teilnehmenden Teams zu verteilen.

Die Vorbereitungsphase, in der die organisatorischen und konzeptionellen Aspekte geklärt wurden, benötigte einen Zeitraum von etwa 5 Monaten (nicht zuletzt durch die räumliche Verteilung der Teilnehmer).

2.2 Architektur

Bei der Konzeption des IDS-Netzwerks wurde auf bestehende Methoden und Lösungen aufgebaut. Das dezentrale Erfassen von Angriffsdaten und eine zentrale Sammlung und Analyse ist ein gemeinsamer Nenner heutiger IDS-Technologie.[1] Die Kommunikation zwischen Sensoren und dem Manager wird bereits weitgehend standardisiert, Spezifikationen der notwendigen Datenformate finden sich beispielsweise in [CDF04]. Entsprechende Sensoren gibt es als hostbasierte Lösung, die auf einzelnen Rechnern selbst detaillierten

[1] Es gibt mittlerweile auch mehrstufige Hierarchien der Analysekomponenten, z. B. mehrere Manager, die nach erster Analyse zusammengefasste Ergebnisse wiederum als Meldungen an einen übergeordneten Manager weiterreichen. Ein entsprechend komplexer Aufbau war für das Generieren der hier beschriebenen Statistiken nicht notwendig und könnte ggf. bei starkem Wachstum des Sensor-Netzes im Nachhinein implementiert werden.

Zugriff auf Systemdaten (z. B. Logmeldungen, Prozess- und Dateisysteminformationen) haben. Es gibt auch netzbasierte Sensoren, die durch Analyse von Netzwerkverkehr in der Lage sind, eine Überwachung von Angriffen für mehrere Rechner eines Netzwerksegmentes vorzunehmen. Da die Statistiken das Gefährdungspotential von vernetzten Systemen illustrieren sollen, war der Einsatz von netzbasierten Sensoren naheliegend, da diese ein sehr weites Spektrum von Angriffen auf Netzdienste durch eine einzelne, relativ einfache Komponente ermöglichen. Da die netzwerkbasierten Sensoren jedoch nur dann in der Lage sind, Angriffe aufzuzeichnen, wenn die Netzwerkanfragen des Angreifers vom Sensor zunächst normal beantwortet werden (z. B. muss der Sensor auf TCP-Verbindungsanfragen tatsächlich die angeforderten Verbindungen aufbauen). Eine entsprechende Lösung findet sich i. d. R. nicht im Bereich der gewöhnlichen IDS-Komponenten.

Die letztlich realisierte und im folgenden diskutierte Architektur des Sensor-Netzwerks besteht aus einer zentralen Server-Komponente sowie beliebig vielen Sensoren (vgl. Abbildung 1).

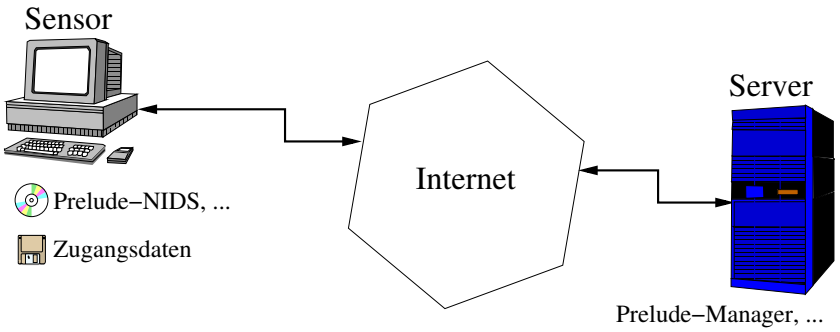

Abbildung 1: Architektur des `eCSIRT.net`-Sensor-Netzwerks

2.3 Anforderungen

Bei der technischen Umsetzung des Sensor-Netzwerks mussten zwei Sichtweisen berücksichtigt werden. Einerseits gibt es starke technische Anforderungen insbesondere auf der Seite der Betreiber, die sich um die Speicherung der Daten und Synchronisation der Sensoren kümmern. Hierzu gehört beispielsweise der verlässliche Betrieb des Managers und eine zu bewältigende Datenmenge an diesem zentralen Sammelpunkt. Dem gegenüber steht die Sicht derjenigen, die Daten erfassen, also einen oder mehrere Sensoren betreiben. Da eine Vielzahl an Sensoren etabliert werden sollte und die einzelnen Betreiber der Sensoren nicht notwendigerweise umfangreiche Kenntnisse im Betrieb von IDS-Sensoren besitzen, musste der Aufwand bei der Inbetriebnahme eines neuen Sensors so gering wie möglich gehalten werden.

Die Anforderungen an ein verteiltes Netzwerk von IDS-Sensoren ergeben sich zum einen direkt aus der Policy (vgl. Abschnitt 2.1), zum anderen aus den generellen Anforderungen

an einen Systementwurf (vgl. z. B. mit [Me90]):

Benutzerfreundlichkeit: Eine einfache Installation neuer Sensoren ermöglicht die problemlose Inbetriebnahme auch durch Personen, die nicht unmittelbar in das Projekt involviert sind oder nur geringe entsprechende Vorkenntnisse besitzen.

Sicherheit: Sicheres Logging garantiert, dass gesammelte Daten nur verschlüsselt und authentifiziert über das Internet übertragen werden. Dies ist nicht zuletzt deshalb von Bedeutung, weil nicht ausgeschlossen werden kann, dass beispielsweise Angriffe aufgezeichnet werden, die aus dem internen Netzwerk des Sensorbetreibers stammen. Außerdem wird das Einschleusen falscher, nicht von den autorisierten Sensoren stammenden Daten verhindert.

Zuverlässigkeit: Das System sollte wartungsarm sein, da prinzipbedingt eine (weiträumige) Verteilung aller Komponenten angestrebt ist. Zudem wird die Wartung mit zunehmender Anzahl der Komponenten immer aufwendiger.

Erweiterbarkeit ist wichtig, damit das System an neue Anforderungen (z. B. Erkennung neuartiger Angriffe) angepasst werden kann.

Skalierbarkeit soll dafür sorgen, dass an möglichst vielen Stellen im Netzwerk Daten gesammelt werden können, ohne dass es beispielsweise Performance-Einbußen o. ä. gibt.

2.4 Implementierung

Die technische Realisierung wurde im Wesentlichen durch frei verfügbare Open-Source-Komponenten realisiert. Die Implementierung bestand also zunächst aus einer Analyse und Auswahl bestehender Komponenten. Danach mussten diese zu einer Gesamtlösung zusammengefügt werden, d. h. das Zusammenspiel musste durch entsprechende Konfiguration und z. T. eigens entwickelte Software ermöglicht werden.

2.4.1 Auswahl der Komponenten

Die Hauptkomponenten des verteilten IDS-Sensor-Netzwerks sind:

- Prelude [Va] als IDS-Framework
- Honeyd [Pr03], um angreifbare Dienste zu simulieren
- NTP mit Kryptographie-Unterstützung [Mi00], um eine Synchronisation der Uhrzeit aller Sensoren zu gewährleisten.
- Knoppix [Kn] als Basis-System für den Betrieb der Sensoren
- Eine für den jeweiligen Sensor angepasste Diskette mit Zugangs- und Konfigurationsdaten sowie entsprechenden Konfigurations- und Startskripten.

Als IDS-Framework kämen u. U. noch AirCERT [PD00] oder M-ICE [Bi] in Frage. Im Falle von AirCERT werden auch IDMEF-Meldungen über einen SSL-Kanal zum Manager übertragen, es wird hierfür jedoch nur ein angepasster Snort-Sensor [Ro99] verwendet. Somit ist nur ein Sensor-Typ möglich, das Framework ist nicht so generisch wie Prelude. M-ICE wurde für den Einsatz von hostbasierten Sensoren (Linux Audit Subsystem, LAuS) entwickelt, netzbasierte Komponenten waren nicht verfügbar.

Prelude erfüllt die wichtigsten der obigen Anforderungen erstaunlich gut. Mit seiner verteilten Architektur ist es flexibel für viele Szenarien verwendbar. Sensoren dienen direkt vor Ort zum Erfassen der Daten und Manager an beliebiger Stelle im Netz zur Aufbewahrung oder zum Weiterleiten („Relaying"), so dass sich sehr komplexe Netzwerktopologien errichten lassen. Ein Sensor kann beliebige Arten von Events aufzeichnen. Beispiele dafür wären netzwerkbasierte Angriffsversuche auf einen Webserver in einem LAN oder fehlgeschlagene Logins auf einem System.

Darüberhinaus unterstützt Prelude das standardisierte Format IDMEF [CDF04], so dass eine Weiterverarbeitung der gesammelten Daten und ein flexibler Ausbau der Sensorfunktionen möglich ist, insbesondere auch mit kommerziellen Produkten, die das IDMEF-Format unterstützen.

Der netzwerkbasierte Sensor von Prelude (Prelude-NIDS) verwendet zum Erkennen von Angriffen die Signaturen von Snort, so dass die ständig aktualisierten Regeln dieses Systems genutzt werden können.

Honeyd ermöglicht, auf einem Rechner vollständige Dienste für Angreifer zu simulieren (inklusive des zugehörigen Netzwerk-Protokollstapels). Alternativ diejenigen Dienste, die für einen potentiellen Angreifer sichtbar sein sollen, tatsächlich zur Verfügung zu stellen, wäre sehr viel aufwendiger in der Konfiguration und vor allem unter Sicherheitsaspekten sehr viel problematischer gewesen.

NTP Version 4 kann mit kryptografischen Methoden sicherstellen, dass die System-Uhren aller Sensoren auf einen vertrauenswürdigen Zeitserver synchronisiert werden. Hierfür wurde der zentrale Server mit einem DCF-77-Empfänger ausgestattet, um die exakte Uhrzeit zu empfangen und an die Sensoren weiterzugeben. Entsprechende Hardware-Uhren für jeden einzelnen Sensor scheiden schon aus Kostengründen als Alternative aus.

Knoppix schließlich ist eine CDROM-basierte Linux-Distribution für x86-kompatible Computer, die sich vor allem durch eine umfassende Hardware-Unterstützung auszeichnet. I. d. R. bootet eine Knoppix-CDROM ohne Probleme auf beliebiger x86-Hardware.

Die Vorteile einer CDROM-basierten Lösung für den Sensor sind, dass ausreichend Speicherplatz für die benötigte Software zur Verfügung steht und dass es kaum Installationsaufwand gibt.

Der Nachteil, dass keine Änderungen an einer CDROM mehr vorgenommen werden können, wird durch die individuelle Diskette für jeden Sensor ausgeglichen. Initial dient die Diskette der Verteilung der nötigen Zugangsdaten (vor allem SSL-Zertifikate und Schlüssel, um dem Prelude-Sensor eine kryptografisch gesicherte Verbindung zu seinem Manager zu ermöglichen). Während der Betriebsphase wird die Diskette gleichfalls benutzt, um Konfigurationsdaten zu speichern.

Da es sich bei bei allen Komponenten um Open-Source-Software handelt, die zudem auf offenen Standards basieren, bietet das System eine sehr flexible Grundlage, um die nötigen Anpassungen vornehmen zu können.

2.4.2 Zusammenfügen der Komponenten

Die Installation des zentralen Servers mit Prelude-Manager, MySQL-Backend und Crypto-NTP-Server war mit vergleichsweise geringem Aufwand zu bewerkstelligen, die Installationen bedurften keiner spezifischen Anpassung.

Für den Betrieb der Sensoren wurde zunächst eine eigene Knoppix-Version erstellt. Dafür wurden im Wesentlichen nicht benötigte Teile entfernt, wie z. B. der X-Server, und stattdessen die Netzwerk-Sensor-Komponente von Prelude (Prelude-NIDS), Crypto-NTP und Honeyd installiert.

Um für jeden Sensor die Möglichkeit zu bieten, individuelle Anpassungen vorzunehmen, wurde zusätzlich auf Basis von Ncurses und Perl ein Konfigurations-Interface entwickelt. Für jeden benötigten Dienst (Prelude-NIDS, Honeyd, Crypto-NTP) können die notwendigen Daten (z. B. IP-Adressen und Netzmasken) eingegeben und rebootfest auf der Diskette gespeichert werden.

Das System wurde mit Hilfe des Linux-Paketfilters Iptables so konfiguriert, dass außer Honeyd und Prelude-NIDS keine Anwendungen IP-Pakete aus dem Internet annehmen können.

Vor allem die letzte Design-Entscheidung sorgt dafür, dass sich die Sensoren vergleichsweise sicher betreiben lassen. Dies ist insbesondere deshalb wichtig, da die installierte Software der Sensoren sich nur vergleichsweise umständlich aktualisieren lässt, ein Update erfordert jeweils das Erzeugen, Verteilen und Brennen von aktuellen CDROM-Images.

3 Betrieb des Sensor-Netzwerks

3.1 Inbetriebnahme

Die initiale Inbetriebnahme des Sensor-Netzwerks erfolgte Ende August bzw. Anfang September. Die ersten vier Sensoren (Deutschland, Niederlande, Großbritannien) gingen innerhalb einer Woche online.

In den ersten Tagen wurden einige Probleme in der Konfiguration der IDS-Signaturen festgestellt. Einige der Netzwerke, in denen Sensoren installiert wurden, benutzen bestimmte Multicast-Protokolle, deren Einsatz vom IDS-Sensor als Angriff aufgefasst wurde. Die häufigen Broadcasts von entsprechenden Paketen führten zu einer großen Anzahl Alarm-Meldungen (Alerts) der IDS-Sensoren. Durch Anpassung der Signaturen wurde dieses Fehlverhalten korrigiert.

3.2 Laufender Betrieb

Der Betrieb des Sensor-Netzwerks über die ersten fünf Monate gestaltete sich sowohl aus Sensor- wie auch aus Server-Sicht sehr wartungs- und störungsarm. Sensor 5 musste aus innerbetrieblichen Gründen vom 31. Oktober 2003 bis zum 7. Januar 2004 offline gehen. Der zentrale Server zum Sammeln der Daten fiel Anfang Oktober aufgrund eines Hardware-Fehlers für einen Tag aus.

Die gesammelten Daten aller Sensoren werden auf dem Server in einer MySQL-Datenbank gesammelt. Momentan (Februar 2004) sind rund 835000 Angriffe in der Datenbank, die eine Größe von knapp 1,8 Gigabytes hat. Die Anzahl der Alerts pro Monat lässt sich Tabelle 1 entnehmen; dabei sind im Oktober auf Grund des Hardware-Ausfalls rund 9000 Datensätze nicht erfasst worden.

Monat	Alerts
September 2003	130349
Oktober 2003	175597
November 2003	178141
Dezember 2003	161134
Januar 2004	125724
Summe	770945

Tabelle 1: Anzahl Alerts aller Sensoren je Monat

Hierbei kann ein einzelner Angriff durchaus mehrere Alerts verursachen, da das Paket des Angreifers von mehreren Signaturen als Angriff erkannt wird. So kann z. B. eine HTTP-Anfrage versuchen, durch Manipulation der Pfad-Angaben aus dem Dokumentenpfad des Webservers auszubrechen, was als erster Angriff erkannt wird. Der versuchte Zugriff auf ein Dokument ausserhalb des Dokumentenpfads kann auf die Datei `/etc/passwd` abzielen, was als zweiter Angriff gemeldet wird. Mehr als zwei Alerts für einen einzelnen Angriff konnten jedoch im praktischen Betrieb bisher nicht beobachtet werden.

Der erzeugte Netzwerkverkehr variiert, da die Sensoren i. d. R. das Alarm auslösende Paket ebenfalls übertragen. Die übertragene Datenmenge je Vorfall hält sich dadurch in Grenzen, dass Prelude eine binäre Kodierung von IDMEF verwendet. Pro Vorfall, der von einem Sensor an den Manager übermittelt wird, fallen durchschnittlich etwa 1250 Bytes IP-Daten an.

4 Veröffentlichung und Auswertung der gesammelten Daten

Um den Teilnehmern Zugriff auf alle gesammelten Daten zu geben, wurde das Webfrontend des IDS-Managers so umgestaltet, dass der Zugriff mittels X.509 Client-Zertifikaten geregelt wurde. Der Zugriff wurde sicherheitshalber nur lesend gestattet, um ein versehentliches Löschen der Datenbank zu verhindern. Über das Webfrontend können die Teilnehmer jeden einzelnen protokollierten Angriff einsehen (einschließlich des gelogg-

ten Netzwerk-Paketes, das den Alarm auslöste) und schon einfache Arten von Statistiken abrufen. Diese Statistiken sind aber reine Mengen-Statistiken; sie zeigen nur die Anzahlen von Alerts an, z. T. sortiert nach verschiedenen Angriffsarten und den verursachenden Rechnern („Top-20-Attackers“). Die Abbildungen 2 und 3 zeigen einige der möglichen Statistiken des Webfrontends.

AttackNb	AttackTypeNb	TargetNb	Score	Address	Country	Host name
15681	12	1	1306.75	**213.131.73.226**	EG	host-213-131-73-226.link.com.eg
6871	17	2	202.088235294118	**194.249.177.100**	SI	n/a
6617	17	1	389.235294117647	**213.136.117.169**	CI	n/a
6613	10	11	60.1181818181818	**127.0.0.1**	Unk.	localhost
6401	17	1	376.529411764706	**192.204.188.101**	US	n/a
6291	17	1	370.058823529412	**194.209.168.106**	CH	n/a
6282	17	1	369.529411764706	**199.6.51.61**	US	n/a
6160	20	2	154	**219.106.233.179**	JP	server2.shirogane.co.jp
5597	12	1	466.416666666667	**212.0.138.6**	SD	n/a
5137	15	3	114.155555555556	**80.108.86.35**	SE	chello080108086035.25.11.vie.surfer.at
4894	5	3	326.266666666667	**210.212.89.4**	IN	n/a
4362	14	1	311.571428571429	**80.142.231.138**	DE	p508EE78A.dip.t-dialin.net
3674	5	2	367.4	**62.117.102.238**	RU	n/a
3346	10	1	334.6	**82.37.219.8**	GB	82-37-219-8.cable.ubr07.dudl.blueyonder.co.uk
3149	17	1	185.235294117647	**211.38.233.233**	KR	n/a
2984	8	4	93.25	**61.189.240.80**	CN	n/a
2532	12	1	211	**217.255.191.239**	DE	pD9FFBFEF.dip.t-dialin.net
2486	11	1	226	**80.142.231.9**	DE	p508EE709.dip.t-dialin.net
2477	12	1	206.416666666667	**80.142.233.137**	DE	p508EE989.dip.t-dialin.net
2460	12	1	205	**217.255.181.26**	DE	pD9FFB51A.dip.t-dialin.net

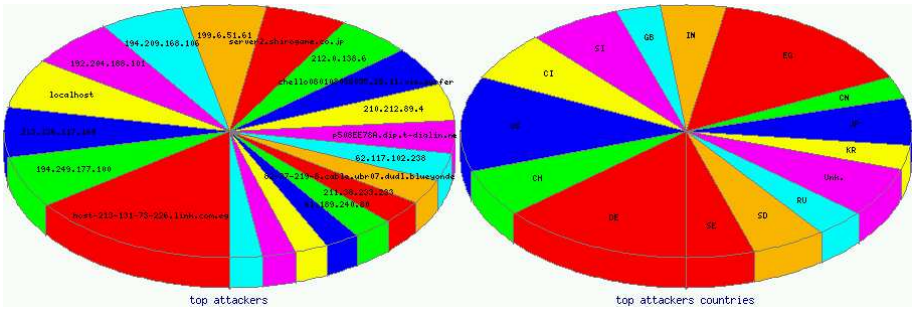

Abbildung 2: Top-20 Angreiferstatistik des Webfrontends [Va]

Für komplexere Statistiken, die insbesondere nicht nur ein einfaches Zählen von Daten, sondern auch eine Korrelation verschiedener Datensätze voraussetzen, stellt das Frontend jedoch keine Funktionalität bereit. Weiterhin basiert das Frontend auf skript-generierten Webseiten, in die dynamisch die gefragten Informationen aus der Datenbank eingefügt werden. Dies führt zum einen dazu, dass die Erzeugung der Statistiken bei jedem Aufruf der Webseiten erneut erfolgt, was bei komlexen Statistiken und großen Datenmengen zu langen Wartezeiten führt. Zum anderen sind die dynamischen Inhalte nicht zur Präsentation auf anderen Webservern und zur Archivierung der Statistiken geeignet. Aus diesem Grunde wurden für die Erzeugung der weiteren Statistiken andere Werkzeuge eingesetzt.

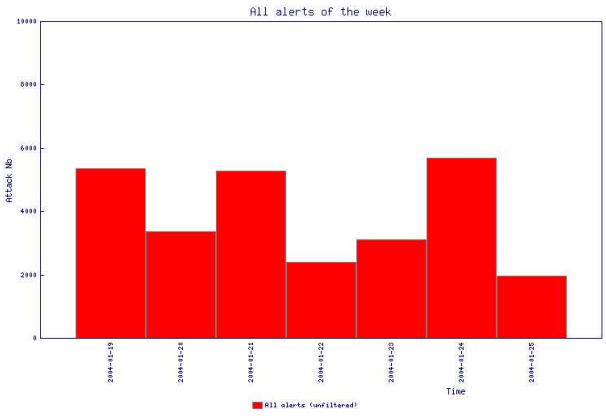

Abbildung 3: Wöchentliche Angriffsstatisitik des Webfrontends [Va]

4.1 Überblick über die Statistiken

Aus allen Vorschlägen für interessante Daten und Zusammenhänge wurden im Projekt zunächst die folgenden ausgewählt:

Gesamtübersicht: Eine Visualisierung der Alerts aller Sensoren über den gesamten Zeitraum.

Monatsübersicht: Anzeige der Anzahl der Alerts aller Sensoren pro Tag, nach Monaten gegliedert.

Tagesübersicht: Anzeige der Anzahl der Alerts aller Sensoren pro Stunde, nach Tagen gegliedert.

Für diese Statistiken wurden die verschiedenen Angriffstypen nach ihren Hauptklassen kategorisiert. Die Hauptklassen wurden nach der durchschnittlichen Häufigkeit der protokollierten Angriffe eingeteilt. Sie bestehen zum überwiegenden Teil aus Angriffen auf WWW-Dienste. Dies liegt zum einen daran, dass die meisten automatischen Angriffe (wie sie z. B. von Würmern verwendet werden) Schwächen in WWW-Servern ausnutzen. Zum anderen können zum aktuellen Zeitpunkt nicht alle möglichen angegriffenen Dienste vollständig von den installierten Honeyd-Skripten simuliert werden. Ein Angriff auf einen SSH-Daemon würde beispielsweise nur in seltenen Fällen erkannt werden, da die Skripte keinen vollständigen SSH-Verbindungsaufbau simulieren, so dass es zum nachfolgenden Angriff gar nicht mehr kommt. Die wesentlichen Kategorien sind zur Zeit die folgenden:

1. `IIS cmd.exe & root.exe access`: Zugriffsversuche auf die genannten Dateien.

2. `HTTP Request String Alerts`: Alle Angriffe auf Webserver, die Fehler in der Interpretation des URL ausnutzen. Dies sind z. B. Versuche, auf übergeordnete Systemverzeichnisse zuzugreifen oder Fehler in der Interpretation des URL-Encodings auszunutzen.

3. `Other IIS Attacks`: Alle anderen Angriffe, die speziell auf einen IIS abzielen.

4. `Other WEB Attacks`: Alle anderen Angriffe, die an Webserver gerichtet sind. Diese betreffen z. B. unerlaubten Zugriff auf verletzliche CGI-Scripte, auf Server-Erweiterungen wie Frontpage und WebDAV.

5. `ICMP Scans`: Alle Versionen von ICMP Echo-Requests.

6. `All others`: Alle anderen Angriffe.

I. d. R. machen die Klassen 1 und 2 etwa 80% der Angriffe aus. Abbildung 4 zeigt exemplarisch die Monatsübersicht für Januar 2004.

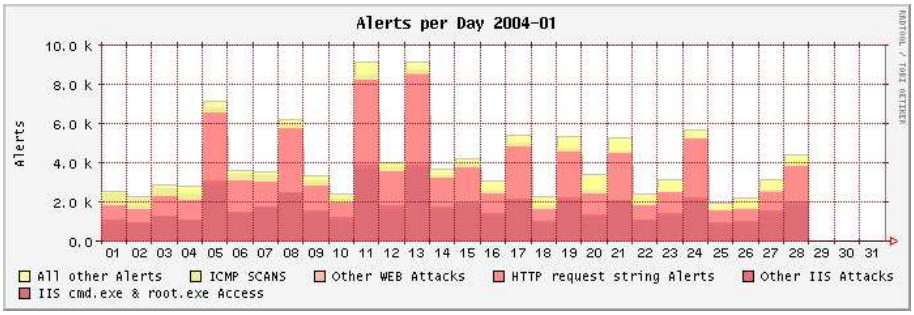

Abbildung 4: Monatsübersicht der Angriffe

Weiterhin wurden die folgenden Statistiken realisiert:

Anzahl verschiedener Angriffe pro Angreifer: Hier wird für jeden Angreifer (identifiziert durch Source-IP-Adresse) ermittelt, wie viele verschiedene Angriffsarten von diesem Host von den Sensoren aufgezeichnet wurden. Dies ermöglicht eine Einschätzung, über welches Repertoire an Angriffen die Angreifer i. d. R. verfügen.

Anzahl der angegriffenen Sensoren pro Angreifer: Diese Statistik nutzt die starke Verteilung der Sensoren auf unterschiedliche Netzwerke aus. Ein Angreifer, der von mehreren Sensoren wahrgenommen wird, versucht offenbar sehr aggressiv, potentielle Opfer im gesamten Internet zu finden. Manche Angreifer haben sich offenbar im Laufe eines Monats an fast allen Sensoren des Netzes versucht, sie scheinen also eine sehr schnelle und aggressive Art zu haben, das Netz zu durchmustern.

Die Tabellen 2 und 3 zeigen die Ergebnisse der ersten fünf Monate.

Angriffe:	1	2	3	4	5	6	7	8	9	10 - 19	20	25	90	132	143
2003-09	13381	2393	1260	107	656	26	11	1	5	7	0	0	0	1	0
2003-10	13547	3311	210	165	581	94	24	15	12	24	2	1	0	0	0
2003-11	14027	3508	50	46	561	42	6	2	2	7	0	0	0	0	1
2003-12	16472	6234	38	32	573	50	5	1	3	12	0	0	0	0	0
2004-01	3595	1239	1363	60	502	51	8	0	1	11	0	0	1	0	0

Tabelle 2: Zahl der Angreifer sortiert nach der Anzahl der verschiedenen Angriffsarten, die sie angewendet haben

Sensoren:	1	2	3	4	5	6
2003-09	17543	267	32	6	0	0
2003-10	17426	360	109	48	42	1
2003-11	17995	211	37	9	0	0
2003-12	23192	175	33	20	0	0
2004-01	6459	319	34	12	7	0

Tabelle 3: Zahl der Angreifer sortiert nach der Anzahl der Sensoren des Netzwerkes, die sie angegriffen haben

Die Anzahl der unterschiedlichen Angriffe der einzelnen Hosts (Tabelle 2) veranschaulicht, dass das Repertoire der Angreifer i. d. R. nicht mehr auf einzelne Exploits beschränkt ist. Drei und mehr Versuche, einen Server unter die eigene Kontrolle zu bringen, werden oftmals angewandt, für gewöhnlich vollautomatisch von Skripten kontrolliert, wie aus den kurzen Zeitintervallen der Angriffe folgerbar ist. Manche Sensoren wurden offenbar auch direkt von Netzwerksicherheitstools unter die Lupe genommen, die wie z. B. Nessus eine umfangreiche Datenbasis an Angriffen verwalten und mehr als hundert verschiedene Alerts auf den Sensoren hervorrufen.

Die Aggressivität, mit der heutige Angreifer in relativ kurzer Zeit einen Grossteil des Internets nach verwundbaren Rechnern absuchen, wird durch Tabelle 3 verdeutlicht. Schon bei einem relativ kleinen, weit verteilten Netz von Sensoren wie dem hier vorgestellten sind einzelne Angreifer sichtbar, die in einem Monat fast alle Sensoren angegriffen haben. Sind diese Fälle offenbar noch recht selten, so sind Angreifer, die mehr als die Hälfte der Sensoren innerhalb eines Monats angegriffen haben, täglich zu beobachten.

Interessant ist auch die folgende Überschlagsrechnung: Ausgehend von etwa 18000 angreifenden Hosts pro Monat sind dies durchschnittlich etwa 2570 Angreifer auf jeden einzelnen Sensor, also etwa 3.5 Angreifer pro Stunde.[2] Man kann also mittlerweile davon ausgehen, dass Systeme, die nicht permanent mit dem Internet verbunden sind (z. B. private Rechner, die sich nur gelegentlich in Internet einwählen) jederzeit gefährdet sind, Ziel eines Angriffs zu sein. Diese Berechnungen auf Basis der exakten Daten aus der Da-

[2]Diese Zahl ist bereits deutlich zu niedrig gegriffen, da nicht zu jedem Zeitpunkt alle sieben Sensoren online waren. Zudem liegt die Anzahl der gezählten Angriffe pro Monat (allerdings inklusive möglicher doppelter Alerts für einen einzelnen Angriff) deutlich über 100000.

tenbank durchzuführen, kann eine weitere sinnvolle Ergänzung der Statistiken darstellen.

Die hier vorgestellten öffentlichen Statistiken [eCb] sind so angelegt, dass sie keine Rückschlüsse auf bestimmte Angriffe auf einen einzelnen Sensor zulassen. Dies stellt einen gewissen Schutz der Betreiber der Sensoren dar. Zusätzlich zu den generischen Statistiken gibt es interne Statistiken, die die Anzahl der Alerts gegliedert nach einzelnen Sensoren zeigen. Die Betreiber von Sensoren erhalten Zugriff auf die gesamten Daten (auch jene von den anderen Sensoren), so dass ein Überblick über die unterschiedlichen Angriffs-Charakteristiken in den einzelnen Netzwerken entsteht und die Erkenntnisse z. B. für die weitere Arbeit in CERTs genutzt werden können.

5 Zusammenfassung und Ausblick

Das erste Resümee, das aus dem Betrieb des verteilten IDS-Sensor-Netzwerks gezogen werden kann, ist durchweg positiv. Der Betrieb des Sensor-Netzwerks lief von Beginn an weitgehend störungs- und wartungsfrei. Aus den gesammelten Daten lassen sich bereits erste interessante Rückschlüsse über das Verhalten von Angreifern gewinnen. Die ermittelten Daten bezogen auf das große Repertoire an Exploits, das einzelne Angreifer verwenden, und die durchschnittliche Anzahl von Angreifern pro Stunde sind durchaus geeignet, den Gefährdungsgrad eines vernetzten Systems zu dokumentieren.

Das Sensor-Netzwerk ermöglicht einen „Blick über den Tellerrand“ des eigenen Netzwerks, da Ereignisse, die an beliebigen Punkten des Internet aufgezeichnet werden, sinnvoll miteinander verknüpft werden können. In Anbetracht zunehmender globaler Bedrohungen des Internet ist diese Herangehensweise nur konsequent und sinnvoll. Quasi als Nebeneffekt wird die Zusammenarbeit aller Partner des Sensor-Netzwerks verbessert, was gerade für Bearbeitung und Bekämpfung sicherheitsrelevanter Vorfälle sehr wichtig ist [Ko01]. Beispielsweise können die gesammelten IDMEF-Daten von Angriffen für die Betreiber der Sensoren durchaus als Datenmaterial für die Vorfallsbearbeitung dienen. Entsprechende Skripte, die die Daten ausgewählter Alerts in Form von IODEF-Objekten (vgl. [DMD]) exportieren, sind bereits in Planung.

Einem weiteren Ausbau des Netzwerks durch Hinzunahme weiterer Sensoren steht nichts im Weg, da die Einrichtung eines Sensors aus technischer Sicht keine Schwierigkeit darstellt. Durch die aktuelle Policy ist die Teilnahme auf akkreditierte Mitglieder von TI beschränkt, diese Beschränkung ist jedoch nur historisch bedingt und wird voraussichtlich in Kürze aufgehoben werden.

Die nächste Version des IDS-Sensor-Netzwerks wird es ermöglichen, IDS-Signaturen im laufenden Betrieb zu aktualisiseren, so dass dann stets die neuesten Angriffe wie z. B. Würmer oder Viren erfasst werden können. Ebenfalls recht weit oben auf der Wunschliste stehen weitere Module für Honeyd, so dass die Sensoren neue Dienste für potentielle Angreifer simulieren können.

Was die Auswertungen angeht, so sind der Fantasie im Grunde keine Grenzen gesetzt, aber ein Hauptschwerpunkt wird darauf liegen, die Daten der verschiedenen Messpunkte noch besser korrelieren zu können. Eine Korrelation der Alerts, die bisher nur auf einer einfa-

chen, signatur-basierten Angriffserkennung basieren, mit anderen Methoden der Datenerfassung und Auswertung ist bereits angedacht. Erste Anstrengungen in dieser Richtung wurden bereits manuell vorgenommen und sind durchaus vielversprechend. Im Idealfall führt dieser Ansatz in Richtung einer netzweiten, frühzeitigen Erkennung neuer Angriffswellen.

Insgesamt ist das bestehende verteilte IDS-Sensor-Netzwerk in seiner jetzigen Form eine gute Grundlage für die geplanten Erweiterungen, da sich das Konzept mittlerweile in der Praxis bewährt hat und dank des Rückgriffs auf Open-Source-Software flexibel ist.

Literatur

[Bi] Biege, T. Modular Intrusion Detection and Countermeasure Environment. `http://m-ice.sourceforge.net/`.

[CDF04] Curry, D. A., Debar, H., und Feinstein, B. Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition. `http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-11.txt`. 2004. IETF internet draft - work in progress.

[DMD] Danyliw, R., Meijer, J., und Demchenko, Y. The Incident Data Exchange Format Data Model and XML Implementation. `http://www.ietf.org/internet-drafts/draft-ietf-inch-iodef-02.txt`. IETF internet draft - work in progress.

[eCa] eCSIRT.net. The European CSIRT Network. `http://www.ecsirt.net/`.

[eCb] eCSIRT.net. `eCSIRT.net` Project – WP4 Public Statistics. `http://www.ecsirt.net/service/ids-sensor-data.html`.

[GDK04] Gellert, O., Dörges, T., und Kossakowski, K.-P.: Understanding Attacks via Distributed IDS. eingereicht für AusCERT2004. 2004.

[Kn] Knopper, K. Knoppix Linux Live CD. `http://knoppix.org/`.

[Ko01] Kossakowski, K.-P.: *Information Technology Incident Response Capabilities*. Books on Demand, Hamburg. 2001. ISBN 3-8311-0059-4.

[KS00] Kossakowski, K.-P. und Stikvoort, D.: A Trusted CSIRT Introducer in Europe. Technical report. Terena – Trans-European Research and Education Networking Association. 2000. `http://www.ti.terena.nl/about_ti/ti-v2.pdf`.

[Me90] Meyer, B.: *Objektorientierte Softwareentwicklung*. Carl Hanser Verlag. 1990. ISBN 3-446-15773-5.

[Mi00] Mills, D. L.: Public Key Cryptography for the Network Time Protocol. Electrical Engineering Report 00-5-1. University of Delaware. 2000. `http://www.eecis.udel.edu/~mills/database/reports/pkey/pkeyb.pdf`.

[PD00] Pickel, J. und Danyliw, R.: Enabling Automated Detection of Security Events that Affect Multiple Administrative Domains. Master's thesis. Carnegie Mellon University, Information Networking Institute. 2000.

[Pr03] Provos, N.: Honeyd – A Virtual Honeypot Daemon. In: Schaumburg, R. und Thorbrügge, M. (Hrsg.), *Sicherheit in vernetzten Systemen. 10. DFN-CERT/PCA Workshop*. DFN-CERT GmbH. Books on Demand GmbH. Feb 2003. ISBN 3-8330-0097-X.

[Ro99] Roesch, M.: Snort – Lightweight Intrusion Detection for Networks. In: *13th Systems Administration Conference (LISA '99) Proceedings*. Usenix Association. Nov. 1999.

[Va] Vandoorselaere, Y. Prelude: An Open Source, Hybrid Intrusion Detection System, Prelude Architecture Guide. `http://www.prelude-ids.org/article.php3?id_article=66`.