Ein Grid-basiertes, föderiertes Intrusion Detection System zur Sicherung der D-Grid Infrastruktur (GIDS)

GIDS – Produktivsystem (MS 36)
Meilenstein zum Abschluss des Arbeitspakets 9

Autoren:
Dr. Wolfgang Hommel (Leibniz-Rechenzentrum)
Dr. Nils gentschen Felde (Ludwig-Maximilians-Universität München)
Felix von Eye (Leibniz-Rechenzentrum)
Jan Kohlrausch (DFN-CERT GmbH)
Matthias Bräck (DFN-CERT GmbH)
Christian Szongott (Regionales Rechenzentrum für Niedersachsen)

GEFÖRDERT VOM

Inhaltsverzeichnis

Kapitel 1

Einleitung

Dieses Dokument präsentiert zusammenfassend die Ergebnisse des Arbeitspakets 9 des Projekts „Ein Grid-basiertes, föderiertes Intrusion Detection System zur Sicherung der D-Grid Infrastruktur“ (GIDS), das die Produktivführung des GIDS-Dienstes als Ziel hat. GIDS (`http://www.grid-ids.de`) ist ein Teilprojekt im Rahmen des D-Grid (`http://www.d-grid.de`) und wird vom Bundesministerium für Bildung und Forschung (BMBF, `http://www.bmbf.de`) gefördert. Weitere Projektinformationen und Unterlagen können der Projekt-Webseite entnommen werden.

1.1 Ziel

Ziel dieses Projekts ist die Bereitstellung eines GIDS-Dienstes für das D-Grid. Hierbei gilt es, soweit wie möglich bestehende Ansätze zu integrieren und ein domänen- und organisationsübergreifendes Gesamtsystem zu entwickeln. Insbesondere die Fähigkeit, mit Virtuellen Organisationen (VO) umzugehen und diese auch als Kunden in Betracht zu ziehen, ist dabei von entscheidender Bedeutung. Die Grundidee ist es, Angriffe durch die kooperative Nutzung und Auswertung von lokalen Sicherheitssystemen zu erkennen. Dazu ist der Austausch von Angriffsdaten und somit deren datenschutzkonforme Aufarbeitung, auch zur Wahrung individuell bestehender Sicherheits- und Informationsverbreitungsrichtlinien, notwendig. In einem kooperativen IDS besteht die Möglichkeit, Angriffe schneller zu erkennen, als dies mit unabhängigen und nur die lokale Sicht berücksichtigenden Sicherheitssystemen möglich ist. Somit kann eine Verkürzung der Reaktionszeit der beteiligten Parteien erzielt werden. Weiter können Vorwarnungen, an zum Zeitpunkt der Erkennung eines Angriffs noch nicht betroffenen Parteien, herausgegeben sowie gegebenfalls präventive Gegenmaßnahmen ergriffen werden.

Eine Auswertung der Daten kann sich zu großen Teilen auf bereits vorhandene Ansätze klassischer IDS stützen. Bei der Auswertung der verfügbaren Datengrundlage ist darauf zu achten, dass VO-spezifische Zugriffsrechte und Befugnisse eingehalten werden. Nach erfolgreicher Auswertung aller verfügbaren Informationen durch ein kooperatives und föderiertes GIDS, unter Beachtung individueller Sicherheits- und Datenschutz-Policies, erfolgt eine Berichterstattung über die erkannten Angriffe auf das Grid oder einzelne beteiligte Partner. Auch hier ist es von Bedeutung, dass eine VO-spezifische Sicht auf die bereitgestellten Informationen realisiert wird. Dazu ist eine Anbindung an die im D-Grid bestehenden VO Managementsysteme zu schaffen. Nach der Entwicklung einer geeigneten Architektur für ein kooperatives und föderiertes IDS in Grid-Umgebungen steht die Implementierung und Produktivführung des Systems. Es soll nach Abschluss der Projektlaufzeit ein produktives Intrusion Detection System als Grid-Dienst im D-Grid zu Verfügung stehen, das sowohl von Ressourcenanbietern als auch von Kunden (VOs, Communities etc.) genutzt werden kann.

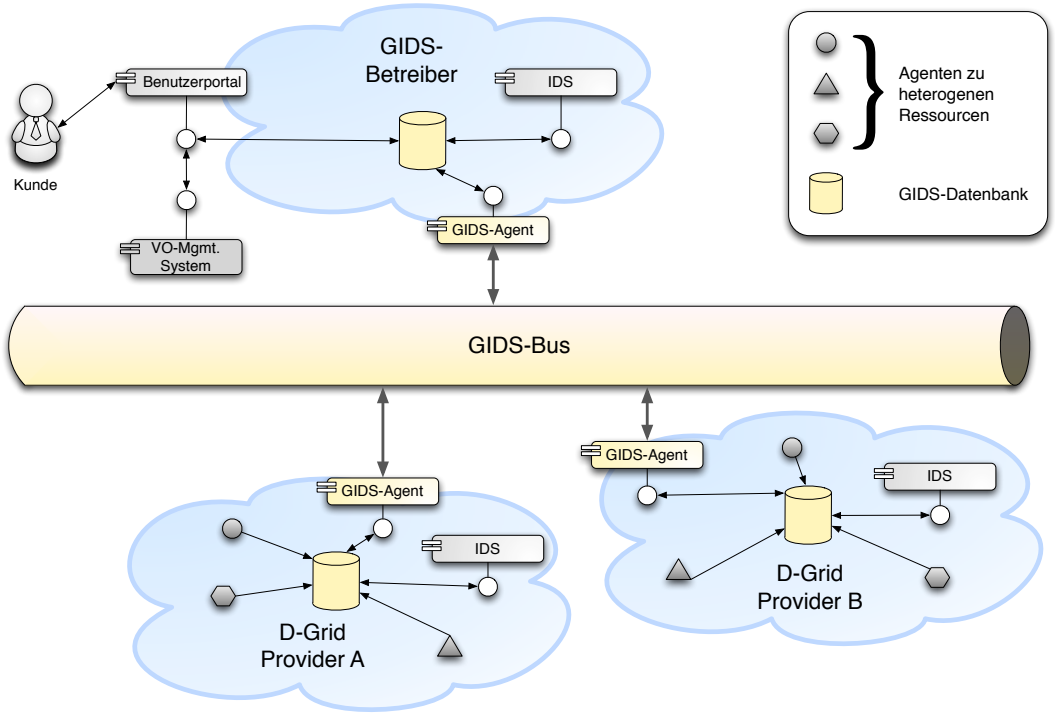


Abbildung 1.1: Überblick über den Aufbau von GIDS

1.2 Lösungsansatz von GIDS

Bereits einleitend in [5, 7, 9] ist die Idee aufgekommen, GIDS als Föderation aus bestehenden, für die Ressourcenanbieter eines Grids spezifischen, sicherheitsrelevanten Komponenten hin zu einem Grid-weiten Frühwarnsystem zu konzipieren. Aus der Analyse in Kapitel 3 der genannten Arbeit [5] sind unter anderem Anforderungen abgeleitet worden, die die Autonomie der einzelnen Teilnehmer eines Grid-basierten IDS fordern, was nicht zuletzt für die Akzeptanz eines solchen Systems zwingend notwendig ist. Daraus abgeleitet bedingt sich eine verteilte Struktur und es entsteht eine lose Kopplung der am GIDS beteiligten Partner, die daraus folgend jeder für sich organisatorisch und administrativ wie auch technisch unabhängig und autonom agieren können und in vielen Bereichen sogar müssen.

Die grundsätzliche Idee von GIDS führt zu einem Grid-globalen Aufbau eines IDS wie es in Abbildung 2.1 wenig detailliert aus einer Vogelperspektive dargestellt ist. Jeder Teilnehmer des GIDS erhält eine zentrale Datenbank, in die alle verfügbaren, für die Sicherheit relevanten Informationen abgelegt werden können. Wie bereits zuvor angesprochen, können dies zum einen Rohdaten (z. B. von versuchten Zugriffen auf gesperrte Ports an einer Firewall) oder auch bereits veredelte oder aggregierte Informationen wie Berichte lokal installierter IDS sein. An einen solchen zentralen Datenspeicher angeschlossen kann ein Agent unter Beachtung einiger notwendiger Randbedingungen Informationen an ein Grid-weites IDS weiterreichen.

1.3 Struktur des Dokuments

Dieses Dokument beschreibt die Produktivführung der prototypischen Implementierung des GIDS, die in [5] vorgestellt wurde. Zuerst wird im Kapitel 2 die Implementierung des Grid-IDS zusammengefasst und es wird auf die Erweiterungen und Anpassungen eingegangen, die im Rahmen der Produktivführung durchgeführt wurden. Dies betrifft beispielsweise die Erweiterungen durch die Kooperation mit dem Frühwarnsystem CarmentiS. Danach wird auf das Datenschutzkonzept des GIDS im Kapitel 3 eingegangen. Für den Betrieb des GIDS sind

verschiedene organisatorische und technische Workflows notwendig, die beispielsweise die Teilnahme am Dienst spezifizieren. Diese werden als Betriebsmodell im Kapitel 4 vorgestellt. Um die Effektivität des GIDS nachzuweisen, werden im folgenden Kapitel 5 die bereits erzielten Ergebnisse beschrieben. Des Weiteren wurde die Ausbreitung mehrerer Internet-Würmer simuliert, um deren Erkennung untersuchen zu können. Abschließend wird in Kapitel 6 in diesem Dokument eine Zusammenfassung der wichtigsten Ergebnisse dargestellt.

Kapitel 2

Implementierung des produktiven GIDS-Dienstes

In diesem Kapitel wird zuerst eine Übersicht über die allgemeine Architektur des GIDS gegeben. Auf dieser Grundlage wird auf die Implementierung des produktiven GIDS eingegangen. Zwar basiert das System und damit der Dienst auf dem Prototypen, jedoch wurde dieser an mehreren Stellen zum produktiven Betrieb weiterentwickelt. Dies betrifft auf der technischen Ebene die Sensorik und das Portal. Eine zusätzliche Erweiterung hat sich aus der Zusammenarbeit mit dem Frühwarnsystem CarmentiS ergeben, die als technische Maßnahme eine Schnittstelle zum Export von Daten nach sich zog.

2.1 Überblick über die GIDS-Architektur

Im Rahmen des Projektes wurde GIDS als Föderation aus bestehenden, für die Ressourcenanbieter eines Grids spezifischen, sicherheitsrelevanten Komponenten hin zu einem Grid-weiten Frühwarnsystem konzipiert.

Die Architektur seitens des Betreibers des GIDS steht in Anlehnung an den Aufbau des Systems auf Seiten eines Ressourcenanbieters. Beides wird in Abb. 2.1 dargestellt. Darüber hinaus stellt der Betreiber einen Grid-Dienst zur Verfügung, der eine Berichterstattung und Darstellung der im GIDS verfügbaren Informationen und Berichte für die unterschiedlichen Nutzergruppen (Ressourcenanbieter, VOs etc.) anbietet. Entsprechend ist die Architektur auf der Seite des GIDS-Betreibers um die Komponenten zur Datenakquise (die Agenten) und die Komponenten zur datenschutzkonformen Informationsaufarbeitung erweitert. Zur Dienstbereitstellung jedoch werden Anbindungen an Grid-typische Dienste (z. B. VO-Managementsysteme) vorgenommen. Im Einzelnen sind die Komponenten:

GIDS-Agent und Datenspeicher. Als wichtigste Aufgabe ist der GIDS-Agent für die Kommunikation mit den anderen Teilnehmern des GIDS verantwortlich. Zum einen verschickt er ausgewählte Informationen (siehe hierzu die Vorverarbeitungsschritte weiter unten) an andere am GIDS teilnehmende GIDS-Agenten, zum anderen empfängt er eben solche Daten von anderen GIDS-Agenten und hinterlegt sie ebenfalls in der zentralen Datenbank.

Sowohl aus Architektur- als auch aus Implementierungssicht ist der GIDS-Agent und der Datenspeicher identisch mit den vergleichbaren Komponenten auf Seiten eines Ressourcenanbieters. Im Falle des Betreibers des GIDS ist es die Hauptaufgabe des GIDS-Agenten, den Datenspeicher mit Informationen, die von den anderen Partnern zur Analyse publiziert werden, zu füllen.

Bevor es zur Veröffentlichung jedweder Information, die lokal bei einem Ressourcenanbieter gewonnen wird, im Grid durch den GIDS-Agenten kommt, durchlaufen sämtliche Informationen noch drei Vorverarbeitungsschritte.

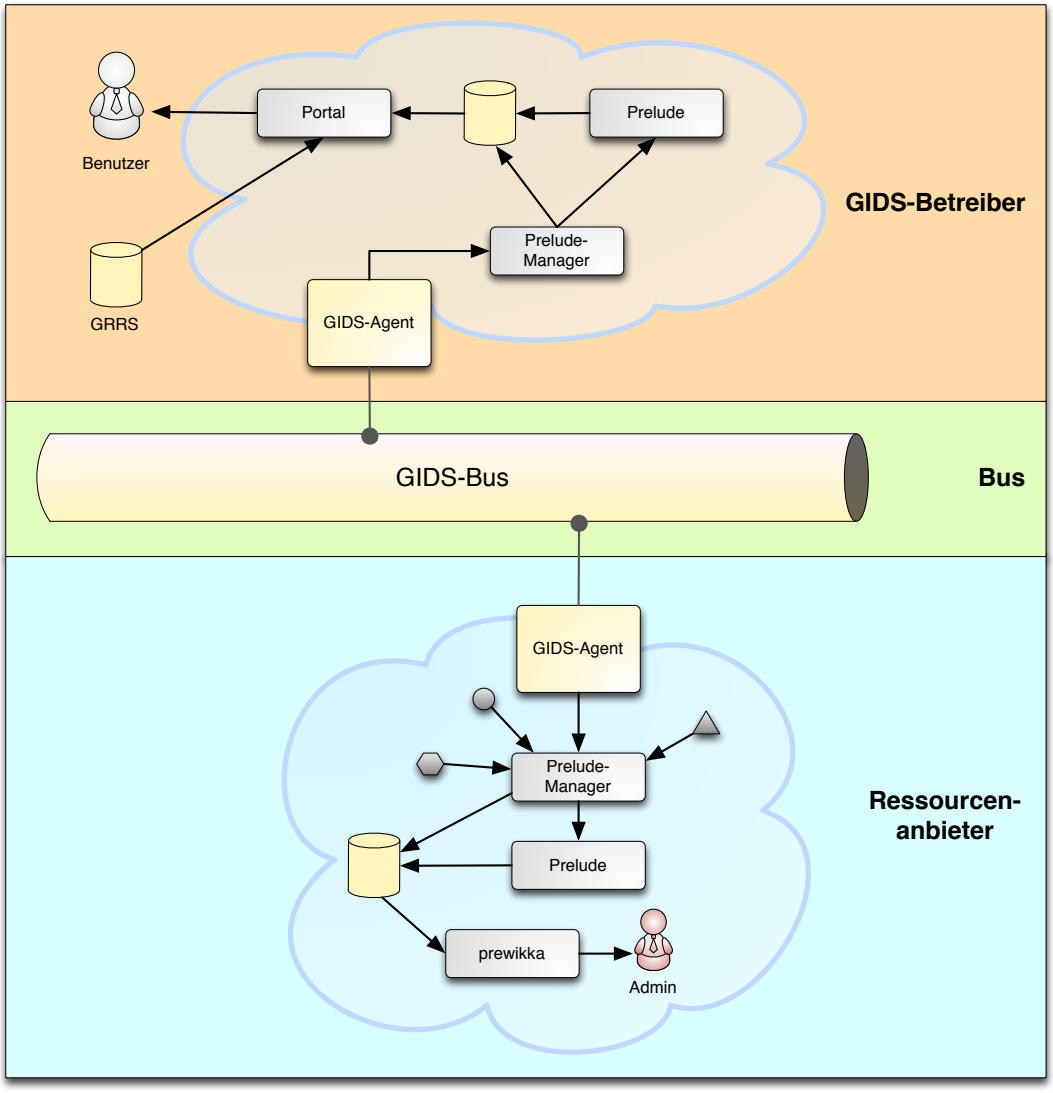


Abbildung 2.1: Überblick über den Aufbau des GIDS-Dienstes

Filter. Neue Datensätze, die in die zentrale Datenbank geschrieben werden und somit zur Weitergabe bzw. Veröffentlichung im GIDS potentiell vorgesehen sind, werden an einen Filter weitergereicht. Die primäre Aufgabe des Filters ist nun das Durchsetzen der Site-spezifischen Informationsverbreitungsrichtlinien. In Abgrenzung zu Datenschutzbestimmungen sind Informationsverbreitungsrichtlinien zumeist Bestandteil lokaler Sicherheitsrichtlinien, die zum Beispiel die Vermeidung der Verbreitung interner Topologiemerkmale, Sicherheitsverletzungen etc. fordern. Der Filter kann einen eingehenden Datensatz auf Grund bestimmter Auswahlkriterien verwerfen oder auch passieren lassen.

Eine weitere wichtige Aufgabe des Filters ist es, Datensätze, die von einem (beliebigen) GIDS-Agenten in die Datenbank geschrieben wurden, auszufiltern. Eine lokale GIDS-Datenbank wird zum einen mit Informationen der bei einem Ressourcenabieter lokal installierten Sensoren gefüllt, zum anderen werden Daten vom GIDS-Bus durch den GIDS-Agenten in die Datenbank geschrieben. Werden nun *alle* neu in die GIDS-Datenbank eingefügten Datensätze wieder veröffentlicht, so wird es zwangsläufig zu Duplikaten in den Datenbeständen und somit zu Endlosschleifen der Nachrichten kommen, wodurch in kürzester Zeit eine Überlastsituation (Netzkapazitäten, Speicherkapazität der Datenbanken, Informationsflut für analysierenden IDS-Instanzen etc.) im GIDS zustande kommen würde.

Aggregator/Verdichter. Diejenigen Datensätze, die nicht zuvor durch den Filter aus dem Informationsstrom entfernt worden sind, können in dieser Komponente aggregiert oder verdichtet werden. Eine Aggregation (auch Konsolidierung oder Verdichtung) bezeichnet das Zusammenfassen vieler Daten mit wenig Informationen zu wenigen Daten mit entsprechend hohem Informationsgehalt. Für eine Aggregation wird eine Aggregationsfunktionen benötigt, die zum Beispiel im Falle einer Menge von Zahlen der Mittelwert, das Minimum, das Maximum oder die Summe sein können. Im Fall eines Grid-basierten IDS bietet es sich an, einzelne, fehlgeschlagene Login-Versuche zu einem Angriff zu aggregieren. Neben der Datenkompression wird der eigentliche Brute-Force- Angriff hervorgehoben, ohne die wesentlichen Informationen zu beeinträchtigen. Durch den Schritt der Aggregation kann also eine Datenverdichtung und Fokussierung erfolgen und somit das Aufkommen und Ungenauigkeiten von Informationen nochmals deutlich gesenkt werden.

Anonymisierer/Pseudonymisierer. Bevor die (aggregierten) Datensätze die administrativen Grenzen eines GIDS-Teilnehmers verlassen, müssen neben den Informationsverbreitungsrichtlinien, die durch die Filterkomponente gewahrt worden sind, auch Datenschutzbestimmungen eingehalten werden. Insbesondere rechtliche Randbedingungen zwingen einen Ressourcenanbieter unter anderem dazu, keine personenbezogenen Daten nach außen zu tragen, was durch den Vorgang der Anonymisierung oder einer Pseudonymisierung gewährleistet wird.

Die Reihenfolge, in der alle Informationen aus der Datenbank die drei zuvorstehenden Komponenten durchlaufen, ist prinzipiell für die Funktionalität nicht entscheidend. Bei der Anordnung dieser Komponenten wird aus Effizienzgründen an erster Stelle eine Filterung (also Löschung „unerwünschter“ Datensätze), dann eine Informationsverdichtung (also eine nochmalige Datenreduktion) und erst abschließend eine Anonymisierung bzw. Pseudonymisierung vorgenommen.

Lokale (G)IDS-Instanz. Dadurch, dass ein (lokaler) Datenbestand sämtlicher im GIDS „öffentlich“ verfügbarer Informationen vorliegt, besteht die Möglichkeit, bei jedem Ressourcenanbieter eine eigene Instanz des GIDS zu betreiben. Dadurch bedingt, dass der Site-spezifische Datenbestand unter anderem auch nicht im GIDS veröffentlichte Informationen enthalten kann, eignet sich diese Instanz des GIDS ebenfalls als mögliche Instanz eines lokalen, Site-spezifischen IDS. Berichte dieses (G)IDS werden ebenfalls in der lokalen Datenbank abgelegt.

2.1.1 Grid-globale IDS-Instanz

Die beim Betreiber des GIDS zum Einsatz kommende Grid-globale IDS-Instanz ist in erster Linie identisch mit einer jeden lokalen (G)IDS-Instanz. In Abbildung 2.1 entsprechen diese Instanzen den Bereichen, die als wolkenartigen Strukturen dargestellt sind. Ein entscheidender Unterschied zwischen den beiden Instanziierungen dieser Komponente ist die Datenbasis, auf der sie arbeiten. Während einer lokalen (G)IDS-Instanz im Wesentlichen die unveränderten Rohdaten der eigenen administrativen Domäne zur Auswertung zur Verfügung stehen, kann die Grid-globale IDS-Instanz ausschließlich auf den anonymisierten und/oder pseudonymisierten Datenbestand der an GIDS beteiligten Partner zurückgreifen.

Die Hauptmotivation für den Betrieb einer Grid-globalen IDS-Instanz ist zweischichtig. Zum einen steht hierbei der Dienstgedanke im Vordergrund. Der Betrieb eines IDS-Dienstes für das D-Grid ermöglicht es Kunden, die keine Ressourcenanbieter sein müssen, einen Überblick über die aktuelle Sicherheitslage im und des D-Grid zu erhalten. Zum anderen stehen natürlich die strategischen Vorteile einer Grid-globalen Ereigniskorrelation und somit einer in einigen Fällen schnelleren oder auch besseren Erkennung verteilt angelegter Angriffe im Vordergrund. Diese Eigenschaft verteilt eingerichteter IDS-Instanzen ist wissenschaftlich unter anderem in [3, 4] belegt.

2.1.2 Benutzerportal

Das Benutzerportal stellt eine zentrale Anlaufstelle für die Kunden des GIDS bereit, in dem kundenspezifische Sichten auf die verfügbaren Reporte realisiert werden. Zur Nutzerauthentifizierung ist eine Anbindung an bestehende AA-Infrastrukturen bzw. VO-Managementsysteme notwendig, die in Abbildung 2.1 als bereits existierende Datenbank GRRS im Grid dargestellt ist. Eine Einschränkung des Nutzerkreises auf Grid-Nutzer ist insofern notwendig, als dass einem externen Angreifer kein Vorteil durch die Einsicht der verfügbaren GIDS-Berichte entstehen soll und er so evtl. Rückschlüsse auf bestehende Sicherheitslücken, Schwachstellen o.ä. schließen könnte.

Alle in diesem Portal verfügbaren Informationen können durch ihre zuvor datenschutzkonforme Aufarbeitung prinzipiell für alle Anwender im Grid einsehbar gemacht werden. Es wird jedoch eine Sicht auf die Berichte angeboten, die nur die eigenen Ressourcen bzw. die von einer VO verwendeten Ressourcen umfasst. Dazu ist eine Abbildung von Ressourcen zu VOs notwendig. Eine solche Datenbank kann z. B. Bestandteil eines Grid-Monitoring-Systems sein wie u. a. im Falle des D-Grid. In jedem Fall kann hier wie bei der Nutzerauthentifikation auf einen bestehenden Dienst im Grid zurückgegriffen werden.

Da das Benutzerportal neben den Abhängigkeiten von bestehenden Grid-Diensten nur von einer GIDS-Datenbank abhängig ist, kann es auch zum Betrieb bei einem beliebigen anderen Teilnehmer des GIDS verwendet werden. Dadurch kann eine redundante Auslegung bzw. eine Wiederverwendung dieser Komponente als Managementoberfläche bei den Ressourcenanbietern ermöglicht werden.

2.2 Implementierung des GIDS-Bus

Die Grundidee von GIDS ist, dass vertrauliche Informationen (mit zum Teil privaten Daten) zu potentiellen Angriffen in einer Föderation (hier innerhalb des D-Grid) über unsichere Medien transportiert werden müssen, um global Hinweise zu Angriffen auszuwerten. Hierfür gilt es eine Lösung zu finden, die einerseits die Sicherheit der Daten gewährleisten und andererseits alle Anforderungen, die sich durch die Infrastruktur selbst ergeben, erfüllen kann.

Über ein *Virtual Private Network* (VPN) können zwei oder mehrere private Netze miteinander verbunden werden. Durch die zusätzliche Verwendung von kryptografischen Verfahren erfolgt die Kommunikation geschützt. Somit kann ein privates Netz auch auf unsicheren Leitungen bereitgestellt werden. In Bezug auf das GIDS-Projekt ist die Realisierung des GIDS-Bus mithilfe dieser Technik denkbar. Zur Nachbildung einer Bus-artigen Kommunikationsstruktur soll eine Multicast-fähige Infrastruktur implementiert werden, die die Vertraulichkeit, Au-

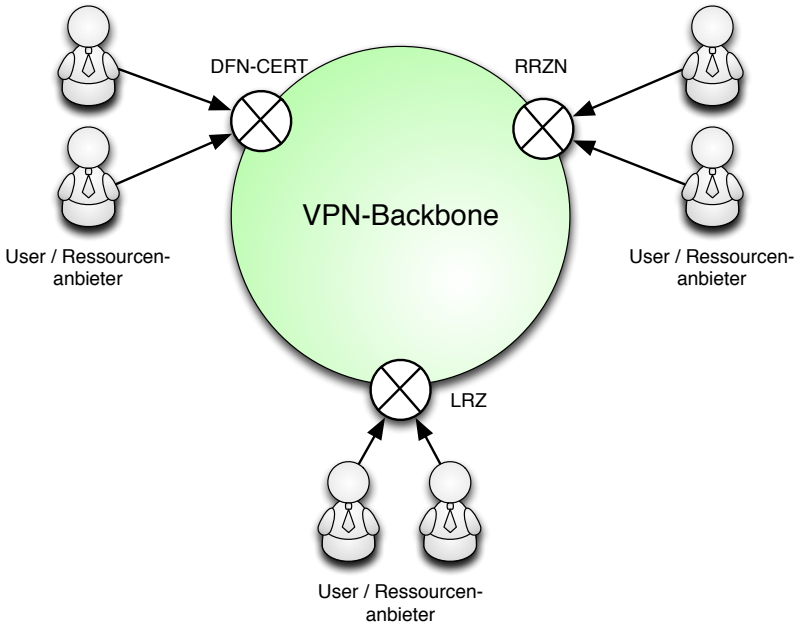


Abbildung 2.2: Grundidee zur Implementierung des GIDS-Bus auf Basis von VPN-Technologien

thentizität und Integrität von übertragenen Daten gewährleistet. Auf Grund der Multicast-Fähigkeit kann dieser Ansatz dann im Weiteren als GIDS-Bus fungieren.

Bei den meisten VPN-Lösungen ist jedoch die redundante Auslegung der Zugangs-Server problematisch. Für eine maximale Ausfallsicherheit im Rahmen von GIDS soll nicht nur eine redundante Auslegung, sondern vielmehr auch eine örtliche und organisatorische Verteilung der Zugangsserver realisiert werden. Den Implementierungsansatz dazu illustriert Abbildung 2.2. Im Rahmen von GIDS ist dieser Ansatz unter Nutzung des freien *OpenVPN* (`http://openvpn.net/`) realisiert.

Der GIDS-Bus wurde in der prototypischen Implementierung getestet und erfüllt die an ihn gestellten Anforderungen (siehe auch [9]). Aus diesem Grund wurde der Bus direkt für den produktiven Dienst übernommen:

Authentifizierung mittels X.509-Zertifikaten Um die Kommunikation durch ein VPN absichern zu können, ist beim Verbindungsaufbau eine gegenseitige Authentifizierung durchzuführen. Sowohl der Server als auch der jeweilige Client müssen sicher davon ausgehen können, dass es sich beim entsprechenden Kommunikationspartner zum einen nicht um einen Angreifer handelt und zum anderen um genau das System, an das die vertraulichen Daten auch übertragen werden sollen. Im D-Grid werden zur Authentifizierung bei der Abgabe von Gridjobs und zur verschlüsselten Kommunikation sogenannte X.509-Zertifikate verwendet. Hierbei handelt es sich um von einer *Certificate Authority* (CA) ausgestellte (signierte) Zertifikate, mit deren Hilfe ein öffentlicher Schlüssel eindeutig einer Entität (Benutzer/Host) im Grid zugeordnet werden kann. Die gesicherte Kommunikation setzt voraus, dass alle im Grid verwendeten Zertifikate von einer CA ausgestellt (also signiert) werden, denen alle Gridressourcen explizit ihr Vertrauen aussprechen. OpenVPN erlaubt zum Aufbau einer authentifizierten Verbindung zum OpenVPN Access Server ebenfalls X.509-Zertifikate. Die bestehende Authentifizierungsstruktur kann also ebenfalls für den GIDS-Dienst verwendet werden. Im Fall eines globalen Diensten kann auf eine eigene CA zurückgegriffen werden, die von der DFN-CERT Services GmbH betrieben wird.

Broadcast- und Multicast-Nachrichten Die sichere Verteilung von Informationen zwischen den GIDS-Agenten und dem GIDS-Provider ist ein wesentlicher Bestandteil der GIDS-Infrastruktur. Informationen müssen nicht nur von den GIDS-Agenten an einen zentralen GIDS-Provider übermittelt werden. Eine wesentlich größere Herausforderung stellt die Übermittlung von allgemeinen Informationen an alle beteiligten GIDS-Agenten dar. Allgemeine Informationen können beispielsweise aus einer Warnung über einen aktuellen Angriff bestehen, der so auf anderen Gridressourcen proaktiv verhindert werden kann. Die Übermittlung von Daten an mehr als einen Empfänger erfolgt mittels OpenVPN über Broadcast-Nachrichten. Dies gewährleistet, dass alle am GIDS-Bus angeschlossenen Agenten die Nachticht simultan empfangen. Für den produktiven Dienst sind dies die Server, die vom Projektkonsortium betrieben werden.

Fallback-Lösung Die am GIDS beteiligten Partner werden über einen OpenVPN Client an den GIDS-Bus angeschlossen. Eine Fallback-Lösung ist für einen produktiven Dienst unumgänglich. Über den Bus werden mehrere, unabhängige OpenVPN Access Server bereitzustellt, welche die Aufgaben des primären Servers nahtlos übernehmen können. Dieser Server müssen, um den gleichzeitigen Ausfall aller Server zu verhindern, auf einer anderen Hardware betrieben werden. Auf diese Weise muss selbst der vorübergehende Verlust einer gesamten Site, z. B. durch Netzprobleme, für das GIDS keine Ausfälle bedeuten. Da auch der Betreiber am GIDS-Bus angeschlossen ist, ist der Import der Daten zum Portal ebenfalls redundant abgesichert.

2.3 Implementierung des GIDS-Agenten

Der GIDS-Agent erfüllt mehrere Aufgaben:

Empfang der Daten Der GIDS-Agent empfängt die Sensor-Daten vom zentralen Prelude-Manager auf der Seite der administrativen Domäne.

Anonymisierung der Daten Der Agent beinhaltet eine Policy zum Filtern und Anonymisieren der IDS-Daten, die von dem Datenschutzkonzept abgeleitet wurde.

Anbindung an den GIDS-Bus Wie bereits vorher beschrieben, erfolgt die Anbindung an den GIDS-Bus über OpenVPN. Die vom Prelude-Manager empfangenden Daten werden vom GIDS-Agent anonymisiert an den Bus weitergeleitet.

Die Sensorik des GIDS-Dienstes und die Kommunikation mit dem GIDS-Bus basiert auf dem *Prelude-Framework*, das als Security Information and Event Management System (SIEM) Lösung für die Sammlung, Auswertung und Weitergabe der Alarmmeldungen auf Seiten der Ressourcenprovider verantwortlich ist. Abbildung 2.3 skizziert den Überblick über die Architektur auf Seiten der Ressourcenprovider. Dabei sind die Komponenten im oberen Drittel des Bildes für die Datenakquise verantwortlich. Die Komponente „export“ ist für den Exportvorgang von IDMEF-Nachrichten aus dem Prelude-Framework verantwortlich.

Implementierung

Die Funktionalität des GIDS-Agenten wurde im Softwarepaket `gids-client` realisiert: Das Paket baut auf der Prelude-Bibliothek libprelude, der Emcast-Bibliothek libemcast und OpenVPN auf, um die Verbindung zum GIDS-Bus und dem Datenexport zu realisieren. Insgesamt besteht das Paket aus den folgenden Komponenten:

OpenVPN Konfiguration Das Paket beinhaltet Skripte und Konfigurationen, um über OpenVPN eine ausfallsichere Verbindung zum GIDS-Bus zu schaffen.

GIDS-Export Die über den Prelude-Manager empfangenen IDS-Meldungen werden über den GIDS-Bus zum Betreiber versendet. Dabei richtet sich der Export nach einer Policy, die die Filterung und Anonymisierung der Daten spezifiziert.

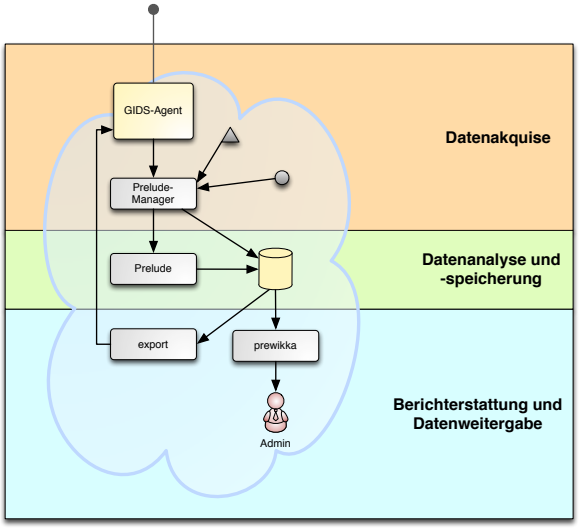


Abbildung 2.3: Überblick über die Komponenten auf Seiten des Ressourcenproviders

GIDS-Import Empfängt Daten vom GIDS-Bus und übergibt diese zur weiteren Verarbeitung an den Prelude-Manager.

Heartbeat Mittels Heartbeats kann sichergestellt werden, dass die Sensorik in Betrieb ist.

Die C-basierte Implementierung des Prelude-Exports verwendet die Bibliothek libxml, um die Verarbeitung des XML-basierte Formats IDMEF zu vereinfachen. Das Programm liest aus dem Prelude-Manager kommende, komplette Meldungen im Prelude-eigenen Binärformat ein. Für jeden Teil, der in dieser Nachricht vorhanden ist, wird durch die entsprechende Funktion der libprelude-API die benötigte Information extrahiert. Zum Schluß wird mit Hilfe der von libxml bereitgestellten Funktionen ein kompletter XML-String im Format IDMEF generiert.

Prelude-Export besitzt zwei Möglichkeiten, um neue IDMEF-Nachrichten zu senden. Entweder werden die Nachrichten in den Dateideskriptor `stdout` geschrieben oder sie werden direkt auf den GIDS-Bus gesendet. Dafür besitzt Prelude-Export eine Anbindung an libemcast, das eine API für die Anwendung Emcast bereitstellt. Welcher Weg für die Dateneingabe benutzt wird, wird in der Konfigurationsdatei `prelude-export.conf` festgelegt. Ist dort `emcast = true` angegeben, wird eine Emcast-Verbindung aufgebaut und alle Nachrichten werden von dort gelesen. In jedem anderen Fall wird `stdout` verwendet.

Um den in [6] beschriebenen und ausgeführten Datenschutzkonzept genüge zu tun, werden alle Meldungen vor einem Exportvorgang gefiltert. Dabei werden alle XML-Knoten, die potentiell den Datenschutz verletzen könnten, gelöscht. Welche Felder das sind oder aus welchen Gründen eine Löschung nötig sein kann, wird in Kapitel 3 beschrieben.

Installation

Im Gegensatz zu anderen Linux Distributionen unterstützt Debian Linux sehr gut das Prelude-Rahmenwerk. Aus diesem Grund wurde die Paketierung des GIDS-Agent für Debian durchgeführt und es wird empfohlen, Debian für den GIDS-Agent zu verwenden. Aufgrund der Flexibilität des Prelude-Rahmenwerkes ist diese keine größere Einschränkung. So kann die Sensorik weiterhin auf den in der administrativen Domäne verwendeten Architekturen betrieben werden. D. h. der GIDS-Agent kann mit jeder anderen Architektur kooperieren. Die Installation erfolgt beispielsweise mittels:

```
dpkg -i gids-client-0.X.x.deb
```

Für Details wird auf die Installationsanleitung des GIDS-Projektes verwiesen.

Konfiguration

Wie schon eingangs erwähnt, gibt es die Konfigurationsdatei `prelude-export.conf`. In dieser kann festlegt werden, ob Nachrichten direkt aus dem GIDS-Bus mittels emcast empfangen werden sollen und was die zugrunde liegenden Parameter dafür sind.

```
#########
# Spezielle Einstellungen für die Verbindung zum GIDS-Bus
#########
[Emcast]
emcast = true        ; Schaltet Emcast-Unterstützung ein.
                     ; Die Alarme werden dann an den GIDS-Bus weitergeleitet.

url = 224.1.2.3:1234 ; Die Broadcastadresse des GIDS-Bus

buffer = 16384       ; Die Größe des Buffers zum Empfang der Daten

loopback = false     ; Wenn Nachrichten, die an den GIDS-Bus versendet
                     ; werden sollen, auch an die absendende Site
                     ; geschickt werden soll, muss hier "true"
                     ; eingetragen werden. ACHTUNG: Diese Einstellung
                     ; kann bei unsachgemäßer Verwendung eine Schleife
                     ; und damit eine Überlastung produzieren!!!
```

Die Angaben im Abschnitt `Anonymisieren` beschreiben die in Kapitel 3 abgedruckte Tabelle. Dabei entsprechen alle mit `D` gekennzeichneten XML-Knoten diejenigen, die potentiell den Datenschutz betreffen und alle mit `LS` gekennzeichneten, die potentiell eine lokale Sicherheitsrichtlinie verletzen. In der Standardkonfiguration werden alle Daten, die den Datenschutz betreffen, ausgesondert, während alle Daten, die potentiell die lokale Sicherheitsrichtlinie verletzen können, weitergegeben werden.

```
[Anonymisieren]
Action.Inhalt = true                    ; D / LS
AdditionalData.Inhalt = true            ; D / LS
Address.address = true                  ; D / LS
Address.netmask = false                 ; LS
Address.vlan-name = false               ; LS
Address.vlan-num = true                 ; D / LS
Analyzer.name = false                   ; LS
Analyzer.manufacturer = false           ; LS
Analyzer.model = false                  ; LS
Analyzer.version = false                ; LS
Analyzer.ostype = false                 ; LS
Analyzer.osversion = false              ; LS
Checksum.key = false                    ; LS
File.name = true                        ; D / LS
File.path = true                        ; D / LS
File.disk-size = false                  ; LS
File.fstype = false                     ; LS
File.file-type = false                  ; LS
FileAccess.Permission = false           ; LS
Impact.Inhalt = true                    ; D / LS
Linkage.name = true                     ; D / LS
Linkage.path = true                     ; D / LS
Node.location = false                   ; LS
Node.name = true                        ; D / LS
OverflowAlert.buffer = true             ; D / LS
```

```
Process.name = false                        ; LS
Process.path = true                         ; D / LS
Process.arg = true                          ; D / LS
Process.env = true                          ; D / LS
Service.name = false                        ; LS
Service.port = false                        ; LS
Service.portlist = false                    ; LS
Service.protocol = false                    ; LS
Service.ip_version = false                  ; LS
Service.iana_protocol_number = false        ; LS
Service.iana_protocol_name = false          ; LS
SNMPService.oid = false                     ; LS
SNMPService.messageProcessingModel = false ; LS
SNMPService.securityModel = false           ; LS
SNMPService.securityName = false            ; LS
SNMPService.securityLevel = false           ; LS
SNMPService.contextName = false             ; LS
SNMPService.contextEngineID = false         ; LS
SNMPService.command = true                  ; D / LS
Source.interface = false                    ; LS
Target.interface = false                    ; LS
UserId.name = true                          ; D / LS
UserId.number = true                        ; D / LS
UserId.tty = false                          ; LS
WebService.arg = true                       ; D / LS
WebService.cgi = false                      ; LS
WebService.url = false                      ; LS
```

Für die Zuordnung einer Meldung zu einer Ressource bzw. zu einer Institution ist es erforderlich, dass die Herkunft der Meldung kenntlich gemacht ist. Für diesen Zweck wird die im D-Grid etablierte GRRS-Datenbank verwendet, bei der eine eindeutige Zuordung einer Grid-Ressource zu einem Verantwortlichen gegeben ist. Weiterhin kann hier angegeben werden, nach wievielen Tagen eine Meldung bei allen anderen Partnern gelöscht werden muss.

```
[SiteSpezifika]
gidsRessource = lrz ; Die gleiche Angabe wie in der GRRS-Datenbank
validUntil = 7      ; Anzahl der Tage, die die Meldung bei anderen Sites
                    ; gespeichert werden kann.
```

Neben diesen globalen Einstellungsmöglichkeiten existieren noch weitere, die Site-lokal von Relevanz sind. Dazu gehört beispielsweise, nach wieviel Sekunden ein Heartbeat gesendet werden soll oder wie die Prelude-Manager-Adresse lautet, von dem die Meldungen empfangen werden sollen.

```
[Prelude]
heartbeat = 600          ; Anzahl der Sekunden,
                         ; nach denen ein Heartbeat geschickt werden soll.
timeout = 1              ; Schlafe eine vorgegebene Anzahl von Millisekunden
                         ; zwischen der Abfrage, ob neue Meldungen im
                         ; Nachrichtenpool vorhanden sind.
manager = 127.0.0.1      ; Die Adresse des Prelude-Managers,
                         ; von dem man die Nachrichten empfängt.
importName = gids-import ; Name des Sensors, der für den
                         ; Importvorgang aus dem GIDS-Bus zuständig ist.
```

Schlussendlich muss Prelude-Export sich noch am zuständigen Prelude-Manager registrieren, damit Prelude-Export zum einen berechtigt ist, Daten mit den Prelude-Manager austauschen

zu dürfen und zum anderen, damit der verschlüsselte Versand vom Meldungen vorbereitet wird. Dazu muss auf der Maschine, auf der Prelude-Export betrieben wird, der Befehl

```
# prelude-admin register prelude-export "idmef:w admin:r"
  localhost --uid 0 --gid 0
Generating 2048 bits RSA private key... This might take a very long time.
```

ausgeführt werden. Dabei muss die Angabe `localhost` durch den Server ersetzt werden, auf dem der Prelude-Manager betrieben wird, mit dem die Verbindung hergestellt werden soll. Weiterhin müssen die Angaben `uid` und `gid` auf die Werte gesetzt werden, die Prelude-Export im laufenden Betrieb hat.

Ist die Generierung des Schlüsselpaares erfolgt, muss am Server des Prelude-Managers die Registrierung abgeschlossen werden. Dazu dient der Befehl

```
# prelude-admin registration-server prelude-manager --listen localhost
Generating 2048 bits RSA private key... This might take a very long time.
```

Auch hier muss die Angabe `localhost` durch den Server ersetzt werden, auf dem der Prelude-Manager betrieben wird. Am Ende wird hier ein Passwort angezeigt, dass auf Seiten der Prelude-Export-Ressource wiederholt muss. Damit ist die Konfiguration von Prelude-Export abgeschlossen.

2.4 Die Sensorik des GIDS-Dienstes

Wie in Abb. 2.3 gezeigt, erfolgt die Datenakquise über mehrere, sich ergänzende IDS-Sensoren. Diese können in Netzwerk-basierte und lokale Sensoren unterschieden werden. Der Vorteil Netzwerk-basierter IDS-Sensoren ist, dass diese aus zentraler Position ein Netzwerk überwachen können. Sie benötigen damit keine speziellen Informationen über die Position und Architektur der Systeme im Netzwerk. In dem GIDS-Dienst wird das IDS Snort für diesen Zweck eingesetzt. Ein weiterer Vorteil von Snort ist die weite Verbreitung und schnelle Erstellung von Signaturen für neue Angriffe und Schwachstellen. Nachteil dieser Klasse von Sensoren ist, dass Angriffe auf dem lokalen System nicht erkannt werden können. Dies betrifft zum Beispiel die Klasse der Schwachstellen, die eine Erweiterung der Privilegien ermöglichen (zum Beispiel „lokaler root Exploit“). Weiterhin können verschlüsselte Verbindungen nicht mit Netzwerk-basierten IDS überwacht werden. Im GIDS übernehmen die IDS Prelude-LML und OSSEC diese Aufgaben.

Das Prelude-Framework baut auf der Bibliothek *libprelude* auf, die Funktionalitäten, wie z. B. den verschlüsselten Transport von Daten zwischen den an libprelude angeschlossenen Komponenten und die Authentifizierung der Komponenten über X.509-Zertifikate, bietet. Diese Funktionalität nutzen alle IDS-Sensoren des GIDS. Allerdings kann die Bibliothek auch in andere IDS eingebunden werden, oder es existieren bereits Erweiterungen für diese. So gibt es bereits eine Unterstützung für das Bro IDS, die die libprelude verwendet. Auf diesem Wege können alternative IDS für das GIDS verwendet werden. Diese werden jedoch zu diesem Zeitpunkt nicht unterstützt.

2.4.1 Snort

Einsatz und Rolle im GIDS

Snort ist ein Netzwerk-basiertes IDS, das in der Default-Konfiguration einen sehr umfangreichen Satz an Signaturen bietet. Signaturen sind sowohl für Angriffe auf Server als auch auf Clients wie beispielsweise Webbrowser vorhanden. Der Einsatzzweck von Snort ist die Erkennung automatisierter Angriffe, die insbesondere die Klasse der Internet-Würmer umfasst. Weiterhin bietet Snort eine gute Erkennung von Angriffen auf die typischen Serveranwendungen wie beispielsweise Web- und Datenbankserver.

Installation und Konfiguration

Das Snort Paket unterstützt in den aktuellen Versionen bereits den Export der Daten durch Prelude. Für die Verwendung von Snort ist es notwendig, den Prelude Export in der Konfiguration zu aktivieren und Snort als Sensor beim Prelude-Manager zu registrieren. Für ein detailierte Beschreibung wird auf die Installationsanleitung des GIDS-Projektes verwiesen. Weiterhin wird vom GIDS-Projekt ein Regelsatz zur Verfügung gestellt, der speziell an Grid-Anwendungen angepasst ist. Da mit dem Prelude-Correlator eine Instanz zur effektiven Reduzierung von Fehlalarmen vorliegt, kann die standardmäßige Konfiguration von Snort übernommen werden.

2.4.2 Prelude-LML

Einsatz und Rolle im GIDS

Prelude-LML ist ein lokaler IDS-Sensor, dessen Regelsatz die Überwachung einer Vielzahl von unterschiedlichen Log-Formaten bietet. Der Einsatz im GIDS ist die Erkennung von Portscans und Brute-Force-Angriffen auf SSH Server. Als wichtiger Vorteil bietet Prelude-LML einen umfangreichen Regelsatz zur Überwachung von Log-Dateien. Dieser umfasst Regeln für eine Vielzahl von Diensten wie beispielsweise SSH- und Webserver. Weiterhin werden die Log-Formate von einer sehr große Anzahl an Firewalls inklusive der kommerziellen Cisco-Produkte unterstützt, was die Überwachung von Firewall Log-Einträgen ermöglicht. Dies ist insbesondere deshalb für das D-Grid wichtig, weil eine Empfehlung für die Konfiguration der Firewall-Regeln im D-Grid existiert, deren Einhaltung einerseits mit dem Prelude-LML verifiziert werden kann und andererseits weisen geblockte Verbindungen auf Angriffe von innen oder außen hin.

2.4.3 OSSEC

Einsatz und Rolle im GIDS

Analog zu Prelude-LML ist OSSEC ein lokaler IDS-Sensor. Neben der Überwachung von Log-Dateien kann OSSEC per kryptografischen Hashes Änderungen bei Systemdateien feststellen. Weiterhin bietet OSSEC ein gute heuristische Erkennung der Aktivitäten von Rootkits. Dies wird durch einen Vergleich der nativen API des Betriebssystem mit anderen Methoden realisiert. Ziel ist es zu erkennen, ob ein Rootkit die API des Betriebssystem so manipuliert hat, dass beispielsweise Dateien oder Prozesse des Angreifers verborgen werden. Dazu vergleicht OSSEC zum Beispiel die Liste der Dateien in einem Verzeichnis, die auf der einen Seite durch die Betriebsystem API und auf der anderen Seite mit einem eigenen Dateisystem-Treiber erstellt wurde. Abweichungen weisen wie bereits vorher beschrieben auf die Aktivitäten eines Rootkits hin.

2.5 GIDS-Portal

Das GIDS-Portal dient als zentrale Anlaufstelle für die GIDS-Partner, die die Gridbenutzer und den Betreiber umfassen, zur Analyse der Ergebnisse. Jeder Benutzer des D-Grids mit gültigem Zertifikat ist berechtigt, auf das Portal zuzugreifen. Jedoch werden aufgrund der derzeit gültigen Datenschutzgesetze nur die Daten angezeigt, für deren Ansicht der Benutzer autorisiert ist. Das für den Betrieb verwendete Portal ist aus dem Prototypen hervorgegangen, wurde aber für einen produktiven Betrieb entsprechend erweitert und verbessert. In diesem Abschnitt wird eine Übersicht über das aktuelle Portal und dessen Erweiterungen für den Betrieb des GIDS gegeben.

Alle Benutzer des D-Grids benötigen ein gültiges Zertifikat zur Nutzung oder Administrierung der Ressourcen. Diese Tatsache macht sich das GIDS-Projekt für die Implementierung der Authentifizierung und Autorisierung für das Portal zunutze. Um nicht-autorisiere Nutzung auszuschliessen, ist das GIDS-Portal daher nur mit einem gültigen D-Grid Gridzertifikat

zugänglich. Im D-Grid sind zwei Zertifizierungsstellen für die Ausstellung von Zertifikaten zuständig:

- DFN-Verein
  `https://www.pki.dfn.de/grid`
- GridKA-CA des KIT
  `http://www.gridka.de/cgi-bin/frame.pl?seite=/ca/d_inhalt.html`

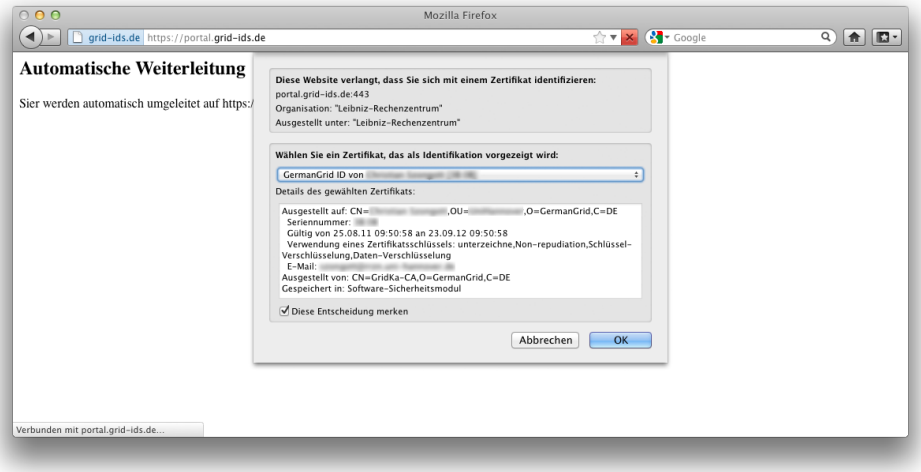


Abbildung 2.4: Authentifizierung mit Hilfe des Gridzertifikats beim Aufruf des GIDS-Portals

Beim Aufruf der URL des GIDS-Portals mit einem Webbrowser wird man dazu aufgefordert, ein entsprechendes Zertifikat vorzuweisen (siehe Abbildung 2.4). Einmal angemeldet erhält der Benutzer die Möglichkeit, sich eine tabellarische Übersicht über die aktuell aufgetretenen Sicherheitsvorfälle und Statistiken über eben jene Alarme anzeigen zu lassen. Zusätzlich können individuelle Einstellungen im Management-Bereich vorgenommen werden.

Alarme Unter dem Menüpunkt „Current Alerts“ gelangen Benutzer zur Anzeige von korrelierten Alarmmeldungen – also Alarmmeldungen, die durch die Korrelation einer Vielzahl einzelner Alarme von Prelude erzeugt wurden (siehe Abbildung 2.5). Die Einschränkung auf korrelierte Alarme hat zum Ziel, den Fokus auf die wichtigen Informationen zu lenken. So ist ein bekanntes Problem bei IDS das Ausblenden von Fehlalarmen (False-Positives). Ein weiteres Ziel ist die Aggregation von Meldungen zur Verbesserung der Übersichtlichkeit. Insbesondere kann beispielweise eine Vielzahl fehlgeschlagener Anmeldeversuche zu einem Alarm („Account-Probe“) zusammengefasst werden.

Die folgenden Felder werden in dieser tabellarischen Darstellung angezeigt:

- **Alert ID:** Eine eindeutige ID der Alarmmeldung des Korrelationsalarms. (Jede einzelne Alarmmeldung, die in einem Korrelationsalarm enthalten ist, besitzt ihrerseits wieder eine eigene eindeutige ID)
- **Analyzer Time:** Ein Zeitstempel, der angibt, zu welchem Zeitpunkt der Analyzer den korrelierten Alarm erkannt hat.
- **Alert Name:** Eine textuelle Beschreibung der Alarmmeldung.
- **Classification:** Alarmmeldungen werden in Kategorien eingeteilt. Der Name der entsprechenden Kategorie wird hier dargestellt.
- **Severity:** Die Severity gibt die Schwere eines korrelierten Alarms an.

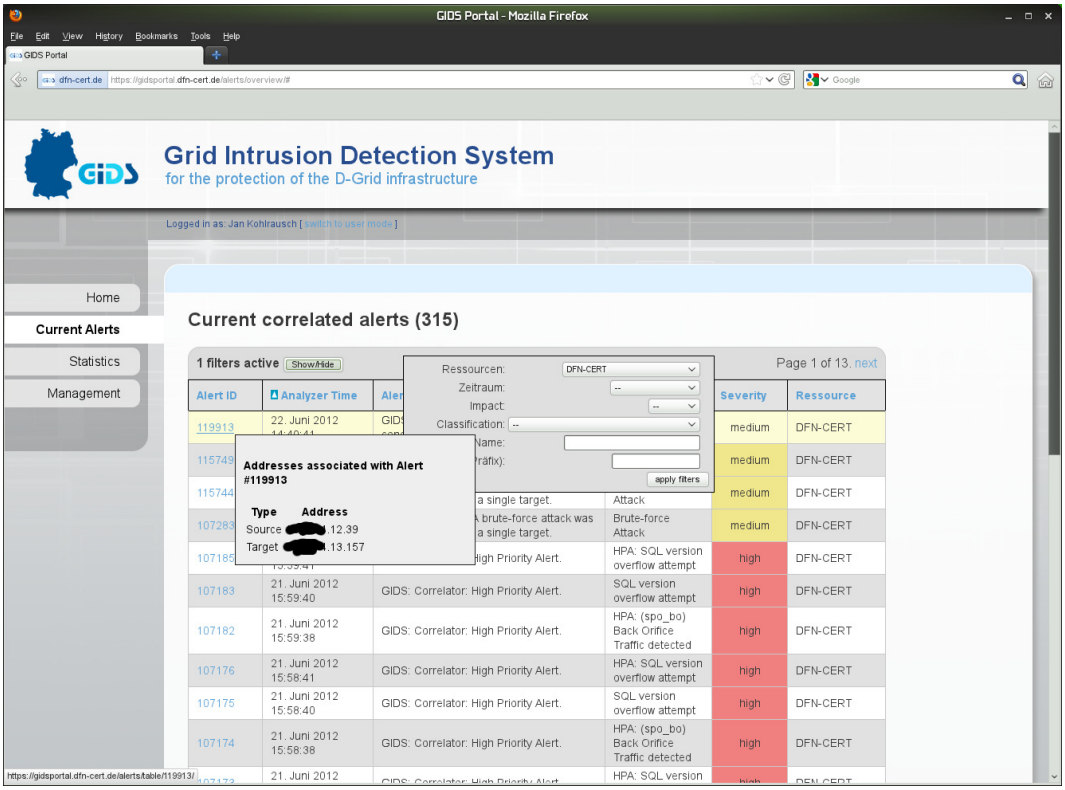


Abbildung 2.5: Tabellarische Darstellung aktueller Korrelationsalarme

- **Ressource:** An dieser Stelle wird angezeigt, auf welche Ressource innerhalb des D-Grids sich die Alarmmeldung bezieht. Dabei wird die ID aus der GRRS-Datenbank verwendet.

An dieser Stelle ist es dem Benutzer möglich, sich nur ausgewählte Alarmmeldungen bestimmter Ressourcen anzeigen zu lassen. Hierfür kann in der entsprechenden Liste eine der Ressourcen ausgewählt werden. Eine zeitliche Filterung der Alarmmeldungen ist ebenfalls vorgesehen. Es kann ein Wert zwischen „allen verfügbaren Meldungen“ bis hin zu Meldungen „der letzten Stunde“ gewählt werden. Weiterhin ist eine Filterung nach der Schwere der Alarme möglich. Um die Übersichtlichkeit der Seite zu wahren, werden pro Seite nur 25 Alarmmeldungen angezeigt. Für das Blättern zwischen den einzelnen Seiten stehen Schaltflächen oberhalb, sowie unterhalb der Tabelle bereit.

Bewegt der Benutzer den Mauszeiger über die ID eines Alarms, so werden involvierte IP-Adressen für den schnellen Überblick eingeblendet. Durch die Auswahl einer Alarmmeldung bzw. den Klick auf eine der angezeigten Alert IDs gelangt der Benutzer zu einer detailliertere Anzeige der ausgewählten Alarmmeldung (siehe Abbildung 2.6). An dieser Stelle werden neben weiteren allgemeinen Informationen zur Alarmmeldungen auch Tabellen mit den entsprechenden Quellen und Zielen des zur Alarmmeldung gehörenden Angriffs aufgelistet.

Innerhalb dieser Tabellen ist es seinerseits wieder möglich, sich detailliertere Information zu jeder einzelnen Quelle bzw. jedem einzelnen Ziel des Angriffs anzeigen zu lassen (siehe Abbildung 2.7).

Hinter dem Punkt „Monitored resources“ verbirgt sich die Möglichkeit, sich eine Liste aller Ressourcen anzeigen zu lassen, auf denen man durch sein Gridzertifikat Zugriffsrechte besitzt. Hier wird neben dem eindeutigen Ressourcenschlüssel (siehe GRRS-Datenbank) das Ressourcen-Kürzel und der vollständige Name der Ressource angezeigt.

Statistiken Die Statistik-Seite soll einen Überblick über den Gesamtzustand („Lagebild“) des D-Grid bieten. Hierfür werden diverse statistische Betrachtungen über die gespeicherten

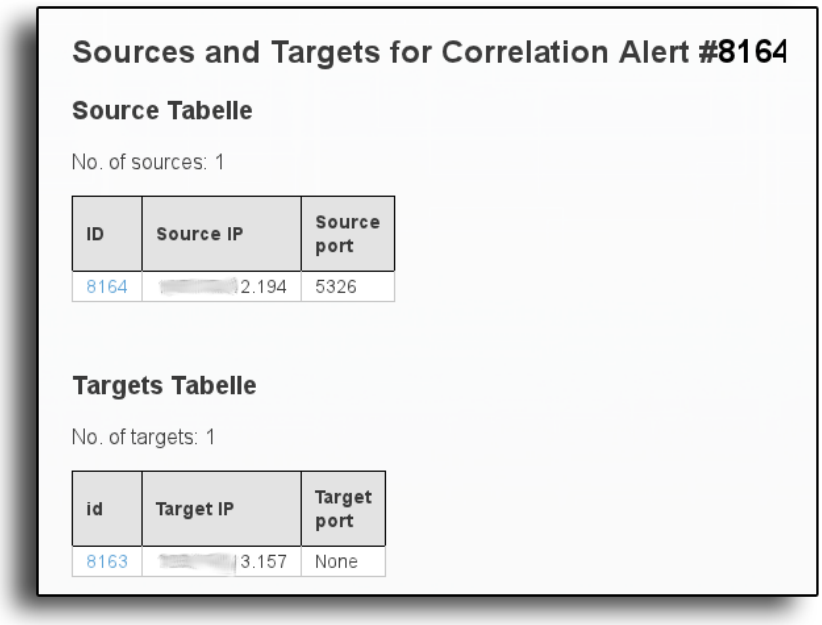


Abbildung 2.6: Detailinformationen zu einer Alarmmeldung

Alarme angewendet und entsprechend grafisch aufbereitet. Dies umfasst die Anzahl der am häufigsten auftretenen Klassifizierungen, Ports und eine Verteilung der Schwere (severity) eines Alarms. Zusätzlich können Benutzer individuelle Statistiken für Ressourcen, auf die sie Zugriff haben, durch entsprechende Auswahl einsehen. Jene Statistiken werden analog zur Gesamtstatistik erzeugt.

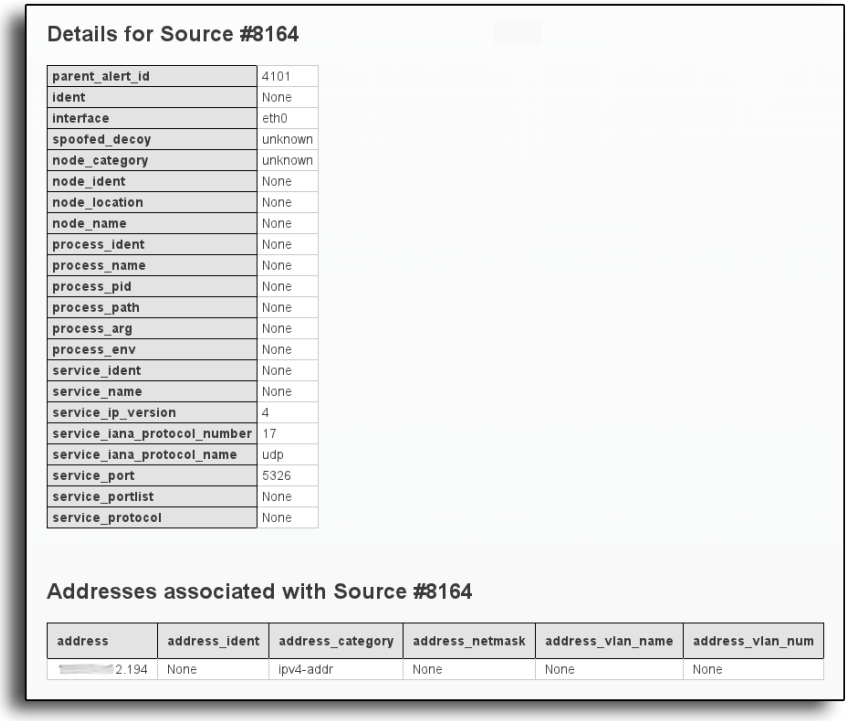
Management Der Management-Bereich (siehe Abbildung 2.8) bietet Ressourcenanbietern und Benutzern die Möglichkeit, diverse Konfigurationen vorzunehmen. So können Ressourcenanbieter einstellen, ob Mitglieder bestimmter VOs der Ressource zugeordnete Alarme nicht einsehen können sollen. Alle Benutzer können außerdem Abonnements anlegen, zu denen automatische Warnmeldung gesendet werden sollen. Eine weitere, benutzerseitige Filterung von Meldungen nach Ressource kann hier ebenfalls vorgenommen werden.

2.5.1 Datenschutz

Das Datenschutzkonzept des GIDS regelt die technischen und organisatorischen Maßnahmen zum Schutz der Daten und bezieht sich auf deren Erhebung, Transport, Verwendung und Speicherung. Für das Portal sind die folgenden Aspekte realisiert worden, um im Einklang mit dem Datenschutzkonzept zu stehen:

Erzwungener verschlüsselter Zugriff auf das GIDS-Portal Aus zweierlei Gründen ist der unverschlüsselte Zugriff auf die von GIDS gesammelten und verarbeiteten Daten zu verhindern.

1. Informationen, die im GIDS-Portal veröffentlicht werden, sind nur für Benutzer des D-Grid bestimmt. Es werden unter Umständen Sicherheitslücken von Ressourcenanbietern aufgezeigt, die auf keinen Fall weiter veröffentlicht werden dürfen. Aus diesem Grund muss der Zugang zum GIDS-Portal unter allen Umständen gesichert erfolgen. Auch Man-in-the-middle-Attacken können auf diese Weise vermieden werden, die die



Details for Source #8164

parent_alert_id	4101
ident	None
interface	eth0
spoofed_decoy	unknown
node_category	unknown
node_ident	None
node_location	None
node_name	None
process_ident	None
process_name	None
process_pid	None
process_path	None
process_arg	None
process_env	None
service_ident	None
service_name	None
service_ip_version	4
service_iana_protocol_number	17
service_iana_protocol_name	udp
service_port	5326
service_portlist	None
service_protocol	None

Addresses associated with Source #8164

address	address_ident	address_category	address_netmask	address_vlan_name	address_vlan_num
[redacted]2.194	None	ipv4-addr	None	None	None

Abbildung 2.7: Detailinformationen zu einer Quelle bzw. einem Ziel eines Angriffs

Veröffentlichung sicherheitskritischer Probleme von Ressourcenanbietern zur Folge haben könnten.

2. Für jede im GIDS-Portal angezeigte Alarmmeldung ist festgelegt, welche D-Grid Benutzer zur Ansicht berechtigt sind. Somit muss die Identität des Benutzers sichergestellt werden, der auf die Seiten des GIDS-Portals zugreift. Durch die Verwendung von X.509-Zertifikaten (Gridzertifikaten) wird nicht nur die Authentifizierung des Benutzers, sondern auch die Autorisierung innerhalb des GIDS-Portals vorgenommen. Der genaue Ablauf wird im folgenden Abschnitt beschrieben.

Die Notwendigkeit zur ausschließlich verschlüsselten Verbindung zum GIDS-Portal hat die folgende zwingende Anforderung an den Webserver zur Folge: Verbindungen über HTTP sind nicht zulässig. Lediglich SSL-gesicherte Verbindungen (HTTPS) sind zu verarbeiten. Um es dem Gridbenutzer ohne Kenntnis dieser Einschränkung zu ermöglichen, das Portal direkt aufzurufen erfolgt beim Aufbau einer ungesicherten Verbindung eine automatische Weiterleitung auf die SSL-gesicherte GIDS-Portal-Seite.

Authentifizierung Wie bereits o. g. darf es nur authentifizierten Benutzer ermöglicht werden, Inhalte des GIDS-Portals abzurufen. In seiner Implementierung wird die Authentifizierung durch den Apache Webserver zusammen mit OpenSSL durchgeführt. Hierbei wird zunächst die Validität des Zertifikats verifiziert. Weiterhin findet eine Überprüfung der Sperrlisten von Zertifikaten statt. Anschließend findet eine Prüfung des Distinguished Names (DN) statt, welcher zwingend in einem X.509-Zertifikat enthalten ist. Befindet sich der DN in einer zu Testzwecken erstellten Whitelist, so wird der Zugang zum Portal gewährt. Andernfalls wird der Verbindungsaufbau abgebrochen und der Zugriff auf das GIDS-Portal bleibt verwehrt.

Durch die Integration in die DFN-CERT Infrastruktur werden auch die hier bereits vorhandenen erweiterten Mechanismen zur Zertifikatsprüfung genutzt. Hierbei wird unter anderem

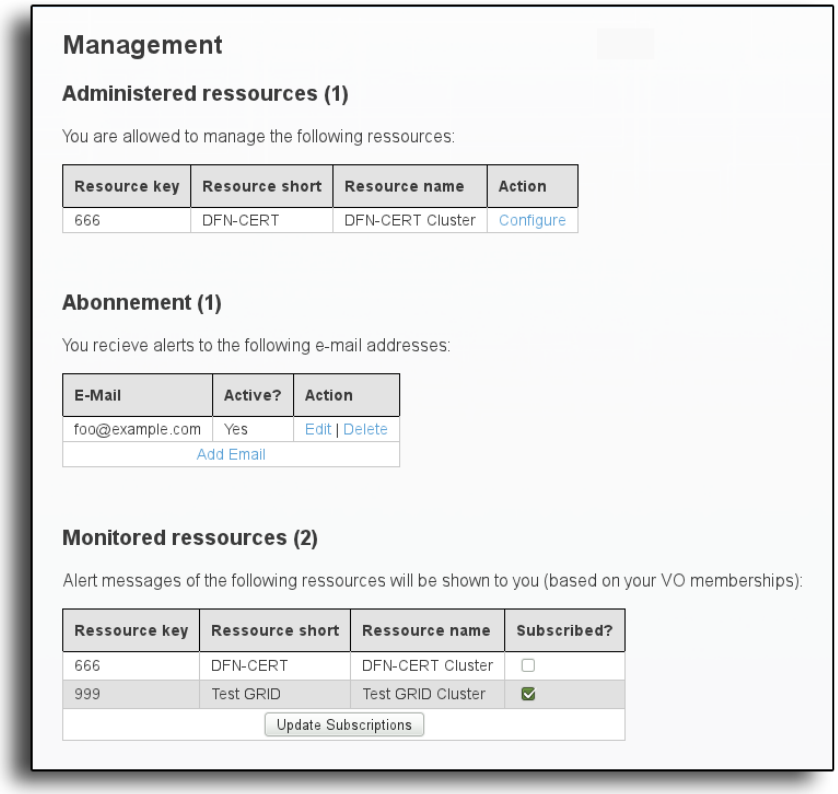


Abbildung 2.8: Management Bereich des GIDS-Portals

geprüft, ob das Zertifikat zurückgerufen (revoked) wurde. Es findet also beim Verbindungsaufbau die Prüfung von Certificate Revocation Liste (CRLs) sowie eine entsprechende Abfrage per Online Certificate Status Protocol (OCSP) statt. Auf diese Weise können auch bereits widerrufene Zertifikate erkannt und abgewiesen werden.

Autorisierung Nach erfolgter Authentifizierung wird die GIDS-Portalseite angezeigt. Der oben bereits erwähnte, im Zertifikat enthaltene DN spielt auch bei der Autorisierung eine entscheidende Rolle. Zur Sicherstellung, dass Gridbenutzer nur Alarmmeldungen von Ressourcen einsehen können, auf denen sie auch die Berechtigung zum Berechnen von Gridjobs besitzen, wird der Zugang zu einer Datenbank des Grid Resource Registration Service (GRRS) benötigt. In zwei Schritten werden diese Ressourcen ermittelt:

1. Im ersten Schritt werden aus der Tabelle `Dgrid_DnVO` passend zum angemeldeten DN des Zertifikats alle VOs ermittelt, in denen der aktuell angemeldete Benutzer Mitglied ist.

2. Im zweiten Schritt werden aus der Tabelle `Dgrid_RessVO` der GRRS-Datenbank alle diejenigen Ressourcen ermittelt, die einer der VOs des Benutzers erlauben, ihre Ressourcen zu nutzen.

Als Ergebnis steht eine Liste aller Ressourcen bereit, die einem Benutzer durch die Mitgliedschaft in VOs zur Verfügung stehen. Anhand dieser Ressourcen-Liste werden in allen Ansichten des GIDS-Portals Alarmmeldungen gefiltert. Die Zuordnung einer Alarmmeldung zu einer Gridressource erfolgt durch ein Ressourcen-Feld innerhalb der GIDS-eigenen Datenbank der Alarmmeldungen.

Abbild der GRRS-Datenbank zur Autorisierung Die GRRS-Datenbank bildet eine grundlegende Informationsquelle über Benutzer, VOs und Ressourcen innerhalb des D-Grids. In seiner grundlegenden Form sind in den Tabellen der Datenbank keine Primär- und Fremdschlüssel vorhanden. Diese werden allerdings zwingend für die Nutzung des Django Frameworks benötigt. Hier werden stets Primärschlüssel in jeder Tabelle benötigt, um eindeutige Python-Objekte erzeugen zu können. Aus diesem Grund befindet sich auf dem Portal-System ein stets aktuelles Abbild der GRRS-Datenbank, welches um die entsprechenden Primärschlüssel erweitert wird. Diese Datenbank wird alle 24 Stunden durch einen automatischen Cronjob mit den aktuellen Datensätzen der originalen GRRS-Datenbank gefüllt. Für die Benutzung des GIDS-Portals entstehen dadurch keinerlei Probleme. Sollte ein neuer D-Grid Benutzer auf das GIDS-Portal zugreifen, kann es zu einer maximal 24-stündigen Verzögerung kommen. Ist ein Benutzer noch nicht im lokalen GRRS-Abbild vorhanden, so wird ihm zwar der Zugang zum GIDS-Portal gewährt. Durch die fehlende Zuordnung zu VOs und entsprechend zu den Ressourcen werden allerdings keine Alarmmeldungen angezeigt.

Filterung Innerhalb des GIDS-Portals kann in der tabellarischen Darstellung, wie in Abschnitt 2.5 beschrieben, eine zeitliche, ein Filterung nach Schwere und auch eine Filterung nach Ressourcen vorgenommen werden. Diese Filterung ist in Django realisiert. Die Abfrage der Daten beim Zugriff auf die GIDS-eigene Datenbank bleibt hierbei unverändert. Die Filterung erfolgt entsprechend der Einstellungen auf der Webseite dynamisch vor der Übergabe an das Webseiten-Template. Darüber hinaus findet vorab eine Filterung nach den getätigten Einstellungen im Management-Bereich statt. Hierbei werden beispielsweise für Mitglieder einer VO, die ein Ressourcenanbieter von seinen Alarmmeldungen ausgeschlossen hat, entsprechende Alarme vorgefiltert.

2.5.2 Implementierung

In diesem Abschnitt wird auf die technische Umsetzung der oben beschriebenen Lösung eingegangen. Da das Portal auf der Version der prototypischen Implementierung basiert, sind die eingesetzten Technologien, Software und Frameworks unverändert geblieben, die zur Implementierung des GIDS-Portals beitragen. Für deren Beschreibung wird auf den Meilenstein [8] verwiesen.

Grundlegende Komponenten Das GIDS-Portal ist in einer eigenen virtuellen Maschine betrieben. Eventuelle Sicherheitslücken in anderen von GIDS verwendeten Komponenten haben so keinen direkten Einfluß auf die Sicherheit und die Funktion des GIDS-Portals. Selbiges gilt entsprechend ebenfalls umgekehrt. Als Basis des Portals dient der Webserver Apache2.[1] Da das GIDS-Portal nach Projektende als Dienst vom DFN-CERT betrieben wird, ist die nahtlose Integration in die bestehende Infrastruktur des DFN-CERTs unumgänglich. Aus diesem Grund wurde auf das Webframework Django[2] zurückgegriffen, welches bereits als Basis für das DFN-CERT Portal eingesetzt wird. Django seinerseits ist ein high-level Python Web Framework, welches die schnelle und komfortable Entwicklung von Webseiten erlaubt und ein Objekt-relationales Mapping zwischen Datenbanktabellen und Python-Objekten ermöglicht. Als weitere Grundlage ergibt sich somit Python als Programmiersprache. Wie bereits o.g. werden innerhalb der GIDS-Infrastruktur eine Vielzahl von Daten in MySQL-Datenbanken gespeichert und verarbeitet. Unter anderem betrifft dies auch die zur Anzeige benötigten Alarmmeldungen. Aus diesem Grund wird ebenfalls auf MySQL-Bibliotheken zurückgegriffen, die von Python in Verbindung mit dem Django-Framework genutzt werden, um aktuelle Datensätze aus entsprechenden Datenbanktabellen abzufragen und aufbereitet im GIDS-Portal anzuzeigen. Zur Darstellung von statistischen Daten wird die JavaScript-Bibliothek flot[3] verwendet, welche eine einfache, grafische Aufbereitung von statischen Daten ermöglicht.

Da bereits einige der Komponenten im DFN-CERT Portal realisiert wurden, sind diese Komponenten in das GIDS-Portal übernommen worden. Das betrifft auf der einen Seite die

[1] The Apache Software Foundation – `http://www.apache.org/`
[2] Django Webframework –`https://www.djangoproject.com/`
[3] flot –`http://code.google.com/p/flot/`

Authentifizierung und Autorisierung von Benutzern, als auch die Zugriffskontrolle der Benutzergruppen auf die Daten. Technisch sind die bestehenden Komponenten aus dem DFN-CERT Portal extrahiert worden und wurden in das GIDS-Portal in Form einer Bibliothek integriert. Im Rahmen der Produktivführung wurden weitere Anpassungen des Portals an die Infrastruktur durchgeführt. Dies betrifft sowohl spezielle Anforderungen an die Netzwerkinfrastruktur als auch die Eingliederung in bestehende Strukturen, wie beispielsweise Schnittstellen zum IT-Frühwarnsystem CarmentiS.

Eine weitere essentielle Anforderung an das GIDS-Portal ist die Sicherung des Zugangs zu den vom GIDS gesammelten und verarbeiteten Daten. Um die verschlüsselte Kommunikation zwischen den Gridbenutzern und dem GIDS-Portal sicherzustellen, wird auf TLS als Verschlüsselungsprotokoll zurückgegriffen. Dem Webserver Apache2 wird über entsprechende Bibliotheken ermöglicht, verschlüsselte Verbindungen zwischen den Browsern der Benutzer und dem GIDS-Portal zur Verfügung zu stellen. Eine weitere Erweiterung betrifft die Prüfung der Gültigkeit von Grid-Zertifikaten. Aufgrund der Verwundbarbeit der PKI-Infrastrukturen, wie beispielsweise den bekannten Problemen mit gestohlenen Benutzer-Zertifikaten, ist dies eine wichtige Anforderung, um derartigen Missbrauch zu verhindern.

2.6 Datenexport nach CarmentiS

CarmentiS ist ein vom BSI und dem CERT-Verbund initiiertes, nationales IT-Frühwarnsystem, das vom DFN-CERT betrieben wird. Im Gegensatz zum GIDS verwendet CarmentiS Angriffsdaten, die von Honeypots und IDS in Sensor-Netzwerken stammen. Da diese Netzwerke nicht produktiv verwendet werden, kann jede Verbindung dorthin als Angriff angesehen werden. Weiterhin lassen sich so größere Netzbereiche überwachen.

Eine der wichtigsten Aufgaben von CarmentiS ist die Erstellung eines Lagebildes. Dafür werden verschiedenen Daten aufgezeichnet und ausgewertet. Das Lagebild gibt einen Überblick über die Gefährdungslage im Internet. Folgende Faktoren werden dafür berücksichtigt:

Internet-Würmer Die bekanntesten Internet-Würmer, wie beispielsweise der W32.Conficker-Wurm, verbreiten sich, indem sie sich selbst-replizierend zu anderen Systemen verbinden und dort versuchen Schwachstellen auszunutzen. Gelingt dies, verbreitet sich der Wurm von diesem neu kompromittierten System weiter. Eine weitere Strategie ist, das lokale Netzwerk nach weiteren verwundbaren Systemen zu scannen. Trotzdem wählen alle Würmer zufällige IP-Adressen aus, um sich über die Grenzen des lokalen Netzes hinaus zu verbreiten. Dies ermöglicht dem CarmentiS System, eine gute statistische Grundlage für die Berechnung der Verbreitung der Internet-Würmer zu erhalten.

Automatisierte Angriffe Eine der größten Bedrohungen im Internet sind durch Bot-Netze gegeben. Diese Netzwerke bestehen aus einer beliebigen Anzahl kompromittierter Systeme, die von einer zentralen Stelle (C&C-Server) gesteuert werden. Eine wichtige Aufgabe der Bot-Netze ist die Suche nach weiteren verwundbaren Systemen. Das kann entweder durch passive Angriffe auf Clients wie beispielsweise Webbrowser („Drive-by Exploit“) geschehen oder durch aktives Scannen. Aktive Scans werden durch CarmentiS erfasst und können hinsichtlich ihre Anzahl ausgewertet werden.

Portscans Angriffe auf schwache Passwörter sind zur Zeit immer noch sehr weit verbreitet. Häufige Ziele sind SSH-, FTP- und Datenbankserver, wie sie auch in Grids verwendet werden. Die Häufigkeit solcher Scans wird auch als Faktor für das Lagebild bei CarmentiS verwendet.

Globale Anomalien Bei CarmentiS wird die Häufigkeit von Verbindungen, getrennt nach der Quelle und Ziel des Angriffs und den entsprechenden TCP- und UDP-Ports ausgewertet. Da Verbindungen in nicht produktive Netze betrachtet werden, kann davon ausgegangen werden, dass es sich bei Verbindungsversuchen um Angriffe und in Ausnahmefällen um fehlkonfigurierte Systeme handelt. Diese Häufigkeiten werden statistisch ausgewertet, um Zustände zu erkennen, die sich von den zu erwartenen statistischen

Schwankungen unterscheiden (Anomalie). Eine Anomalie kann beispielsweise durch einen neuartigen Internet-Wurm oder eine neue, aktuell ausgenutzte Schwachstelle entstehen.

Neben den Ergebnissen der eingesetzten Sensorik fließen auch weitere Informationen in das Lagebild ein. Dies sind die Einschätzungen anderer Teams im kommerziellen oder akademischen Umfeld, beispielsweise Hersteller von Anti-Virus Software, andere CERTs und Sicherheitsteams.

Auf der einen Seite kann GIDS von dem Lagebild in CarmentiS profitieren. Das Lagebild gibt wichtige Bedrohungen wider, die auch Grids betreffen oder Auswirkungen auf diese haben. Weiterhin ist eine Korrelation mit Daten aus CarmentiS möglich. So kann potentiell bei einer Verbindung zu einem Grid System überprüft werden, ob von diesem System andere Angriffe ausgegangen sind. Das ist beispielsweise bei einem Anmeldevorgang nützlich, bei dem unklar ist, ob gestohlene Passwörter oder Zertifikate ausgenutzt wurden. Auf der anderen Seite kann CarmentiS von den Angriffsdaten des GIDS profitieren. Da die Daten von CarmentiS auch zur Warnung der betroffenen Seite durch das Incident Response Team (IRT) des DFN-CERTs verwendet werden, bietet sich das GIDS als weitere Datenquelle an.

Export der Daten Der Export der Daten erfolgt analog zu dem Export der Daten durch den GIDS-Client. Anstelle des GIDS-Bus werden die Daten an die entsprechenden Schnittstellen von CarmentiS übergeben. Dazu wurden entsprechende Schnittstellen auf der Seite des Betreibers in das GIDS eingefügt, die die Daten für CarmentiS aufbereiten und in das geeignete Datenformat überführen. Technisch hat dies den Vorteil, dass diese Lösung den Export von allen am GIDS beteiligten Partnern ermöglicht. Willigt ein Partner dem Datenexport zu, werden dessen Daten automatisch nach CarmentiS exportiert, ohne dass der Partner den Export manuell durchführen muss und somit keinerlei weitere Aufwände bei diesem entstehen.

Allerdings muss beachtet werden, dass im GIDS auch IDS-Meldungen auftreten können, die keine Angriffe wiedergeben. Da bei CarmentiS von Angriffsdaten ausgegangen wird, müssen diese Daten vor dem Export herausgefiltert werden. Dies geschieht durch Filter, die sowohl die Quelle als auch die „Severity“ (Schweregrad) der IDS-Meldung berücksichtigen.

Kapitel 3

Datenschutz

Im Projekt GIDS werden sicherheitsrelevante Informationen innerhalb einer geschlossenen Nutzergruppe im D-Grid zum Zwecke einer verbesserten Angriffserkennung ausgetauscht. Zum Schutz dieser Daten wurde im Rahmen des GIDS-Projektes ein Datenschutzkonzept erarbeitet und auf das IDMEF Datenaustauschformat angewendet. Dabei spielen zwei Klassen von Daten eine wichtige Rolle:

Personenbezogene Daten sind durch das Bundesdatenschutzgesetz bezüglich deren Erhebung, Verwendung, Transport und Speicherung geschützt. Dabei wird zwischen den Daten unterschieden, die direkt auf eine Person hinweisen (bestimmt) oder mittels zusätzlichem Wissen auf eine Person schliessen lassen (bestimmbar).

Sicherheitskritische Daten geben Informationen preis, die von den GIDS-Partnern oder dem Betreiber als sicherheitskritisch eingestuft werden. Zum Beispiel sind das Informationen über die IDS-Sensorik, die von einem Angreifer zur Verschleierung von Angriffen missbraucht werden können.

Dieses Datenschutzkonzept betrifft im Wesentlichen die Weitergabe und Verarbeitung der IDS-Sensordaten. Jedoch betrifft es auch den Zugriff und die Speicherung der Daten im GIDS-Portal.

3.1 Umsetzung der Datenschutzrichtlinie

Zur Einhaltung des Datenschutzgesetzes gibt es zwei Alternativen, die entweder die Vermeidung personenbezogener Daten oder die Schaffung einer gesetzlichen Grundlage als Ziel haben. Die Vermeidung kann technisch durch die Anonymisierung der Daten realisiert werden. Dafür ist ausreichend, dass die Daten keinen Bezug zu genau einer Person zulassen. Als Nachteil der Anonymisierung ist die Korrelation der Daten nicht mehr möglich. Zwar kann dieser Nachteil durch Pseudonymisierung vermieden werden, jedoch ist streng genommen der Bezug zu der entsprechenden Person immer noch möglich. Vorteil ist die einfache technische Realisierung. Die Alternative ist, eine gesetzliche Norm zur berechtigten Nutzung personenbezogener Daten zu schaffen. Dies kann beispielsweise durch entsprechende Verträge zwischen den GIDS-Partnern und dem Betreiber realisiert werden, in denen dem Betreiber das Recht eingeräumt wird, die Daten zu verarbeiten (Auftragsdatenverarbeitung). Weiterhin kann argumentiert werden, dass die Verarbeitung dieser Daten zur Beseitigung von Störungen durch Angriffe notwendig ist.

Während der Projektlaufzeit fiel die Entscheidung für die Anonymisierung personenbezogener Daten. Der entscheidende Vorteil dieser Lösung ist, dass keine vertragliche Bindung notwendig ist. Die Erfahrung mit ähnlichen Diensten hat gezeigt, dass dies die Schwelle des Beitritts und die Akzeptanz deutlich erhöht. Deshalb wird diese Variante auch für den Start des GIDS-Dienstes gewählt. Da die technischen Möglichkeiten die Verarbeitung der nicht-anonymisierten Daten vorhanden ist, kann auf diese Alternative später umgeschwenkt werden; zum Beispiel, wenn die Akzeptanz des GIDS-Dienstes sichergestellt ist.

3.1.1 Löschung innerhalb einer Seite auf Attributebene

Um einen maximalen Datenschutz gewährleisten zu können, werden in der Standardkonfiguration alle mit (d) markierten Felder komplett gelöscht. Dies vermindert zwar die Einstiegshürde für neue Teilnehmer des GIDS, verhindert aber eine sinnvolle Korrelation der Daten. Gerade für IP-Adressen wurde daher eine alternative Anonymisierung implementiert, die das letzte Byte konstant auf Null setzt. Diese Maßnahme genügt ebenfalls dem Datenschutz und bietet zusätzlich die Möglichkeit der Korrelation. Da diese Form der Anonymisierung jedoch zum Teil Rückschlüsse auf die Topologie der Ressourcen-Provider erlaubt und dies die Einstiegshürde erhöhen würde, wurde in der Standardkonfiguration darauf verzichtet, es kann jedoch jederzeit umkonfiguriert werden. Hat sich eine hohe Akzeptanz des Dienstes gebildet, kann im zweiten Schritt die Anonylisierung durch eine Pseudonymisierung (beispielsweise CryptoPAn) oder eine weniger restrikte ersetzt werden.

Die mit (ls) markierten Attribute werden in der Defaulteinstellung nicht verändert. Welche von diesen Attributen gelöscht werden, müssen die Ressourcenprovider selbst treffen. Im Programm, dass innerhalb des GIDS-Projektes entwickelt wurde, um die Alarme an alle anderen Sites schicken zu können, werden den Administratoren Konfigurationsdateien zur Verfügung gestellt, mit denen sie eine Löschung der Attribute einfach durchführen können.

Wichtig ist dabei anzumerken, dass die Sensorenbetreiber selber dafür verantwortlich sind, ob lokale Sicherheitsrichtlinien oder der Datenschutz verletzt sind. Somit können noch weitere, hier nicht vermerkte Felder kritisch in Bezug einer Weitergabe sein, auf der anderen Seite können auch hier markierte Felder im Kontext eines Ressorcenproviders völlig harmlos sein und problemlos weitergegeben werden. In der angesprochenen Konfigurationsdatei ist eine feingranulare Löschung möglich.

3.1.2 Datengrundlage/Filterung ganzer Alarmmeldungen

Gemäß dem Grundsatz, dass nicht gesammelte Daten diejenigen Daten sind, die nach Datenschutzaspekten am Besten geschützt sind, sollen im GIDS-Projekt nur Daten ausgetauscht werden, die im direkten Zusammenhang mit Grid-Ressourcen stehen. Es liegt somit im Verantwortungsbereich der Ressourcenprovider, alle nicht Gridressourcen betreffende Alarme auszufiltern. Weiterhin besteht die Möglichkeit, ganze Alarmmeldungen auszufiltern, falls diese der Zielsetzung von GIDS entgegensprechen oder es lokale Sicherheitsbedenken gegen eine Weitergabe bestehen.

3.1.3 Löschung externer Alarmmeldungen

Wird eine Alarmmeldung an andere Sites verschickt, so wird diese beim Absenden mit einem Gültigkeitsdatum versehen. Dieses entspricht der Gültigkeitsdauer der Alarmmeldung bei der lokalen Site, das heißt, einer vom Erstellungszeitpunkt aus gerechnet den lokalen Datenschutzrichtlinien entsprechende Anzahl von Tagen. Wird nichts vorgegeben, so wird diese Anzahl von Tagen den aktuellen Gerichtsentscheiden entsprechend auf sieben Tage gesetzt. Es ist jedoch im Ermessensspielraum eines jeden Ressorcenproviders, diese Frist gegebenenfalls zu erhöhen oder zu verringern. Die Gültigkeitsdauer von Alarmen externer Sites ist jedoch unabhängig von einer solchen Änderung.

Bei jeder Site wird ein Cronjob installiert, der die Einhaltung dieser Löschung übernimmt. Mit Teilnahme am GIDS-Netzwerk erklärt sich jede Site damit einverstanden, dass sie dafür verantwortlich sind, dass dieser Cronjob seinen Dienst täglich wenigstens einmal durchführen kann.

3.2 Anwendung auf den IDMEF Standard

Welche sicherheitsrelevanten Informationen in einer IDMEF-Nachricht versendet werden beziehungsweise welche Teile innerhalb einer Nachricht weitergegeben werden, ist Inhalt dieses Abschnitts. Dazu wird der Standard IDMEF kurz vorgestellt und an wichtigen Stellen, an den

es aus datenschutzrechtlicher Hinsicht Bedenken geben könnte oder die Daten zu sensibel sind, um sie weiterzugeben.

Im Folgenden sind zur Übersicht alle im IDMEF-Standard verwendeten Klassen aufgeführt. Dabei bedeutet ✔in der Spalte *D*, dass dieses IDMEF-Feld den Datenschutz erfüllt und ✔in der Spalte *LS*, dass lokale Sicherheitsrichtlinen nicht tangiert werden. Analog dazu bedeutet ein ✘in der Spalte *D*, dass dieses IDMEF-Feld den Datenschutz potentiell verletzen könnte und ✘in der Spalte *LS*, dass lokale Sicherheitsrichtlinen verletzt werden könnten.

Klasse	**Attribut oder Unterklasse**	**D**	**LS**
Action (3.3.13)	category	✔	✔
	Inhalt	✘	✘
AdditionalData (3.3.11)	meaning	✔	✔
	type	✔	✔
	Inhalt	✘	✘
Address (3.3.17)	address	✘	✘
	category	✔	✔
	ident	✔	✔
	netmask	✔	✘
	vlan-name	✔	✘
	vlan-num	✘	✘
Alert (3.3.1)	AdditionalData	✔	✔
	Analyzer	✔	✔
	Assessment	✔	✔
	AnalyzerTime	✔	✔
	Classification	✔	✔
	CreateTime	✔	✔
	DetectTime	✔	✔
	messageid	✔	✔
	Source	✔	✔
	Target	✔	✔
Analyzer (3.3.6)	Analyzer	✔	✔
	analyzerid	✔	✔
	class	✔	✔
	manufacturer	✔	✘
	model	✔	✘
	name	✔	✘
	Node	✔	✔
	ostype	✔	✘
	osversion	✔	✘
	Process	✔	✔
	version	✔	✘
Assessment (3.3.10)	Action	✔	✔
	Confidence	✔	✔
	Impact	✔	✔
Checksum (3.3.28)	algorithm	✔	✔
	key	✔	✘
	value	✔	✔
Classification (3.3.7)	ident	✔	✔
	Reference	✔	✔
	text	✔	✔
Confidence (3.3.14)	rating	✔	✔
	Inhalt	✔	✔

Klasse	**Attribut oder Unterklasse**	**D**	**LS**
CorrelationAlert (3.3.3)	alertident	✔	✔
	name	✔	✔
File (3.3.24)	access-time	✔	✔
	category	✔	✔
	Checksum	✔	✔
	create-time	✔	✔
	data-size	✔	✔
	disk-size	✔	✘
	FileAccess	✔	✔
	file-type	✔	✘
	fstype	✔	✘
	ident	✔	✔
	Inode	✔	✔
	Linkage	✔	✔
	modify-time	✔	✔
	name	✘	✘
	path	✘	✘
FileAccess (3.3.25)	Permission	✔	✘
	UserId	✔	✔
Heartbeat (3.3.5)	AdditionalData	✔	✔
	Analyzer	✔	✔
	AnalyzerTime	✔	✔
	CreateTime	✔	✔
	HeartbeatIntervall	✔	✔
	messageid	✔	✔
Impact (3.3.12)	completion	✔	✔
	severity	✔	✔
	type	✔	✔
	Inhalt	✘	✘
Inode (3.3.27)	change-time	✔	✔
	c-major-device	✔	✔
	c-minor-device	✔	✔
	major-device	✔	✔
	minor-device	✔	✔
	number	✔	✔
Linkage (3.3.26)	category	✔	✔
	File	✔	✔
	name	✘	✘
	path	✘	✘
Node (3.3.16)	Address	✔	✔
	category	✔	✔
	ident	✔	✔
	location	✔	✘
	name	✘	✘
OverflowAlert (3.3.4)	buffer	✘	✘
	program	✔	✔
	size	✔	✔

Klasse	Attribut oder Unterklasse	D	LS
Process (3.3.20)	arg	✘	✘
	env	✘	✘
	ident	✔	✔
	name	✔	✘
	path	✘	✘
	pid	✔	✔
Reference (3.3.15)	meaning	✔	✔
	name	✔	✔
	orgin	✔	✔
	url	✔	✔
Service (3.3.21)	iana_protocol_name	✔	✘
	iana_protocol_number	✔	✘
	ident	✔	✔
	ip_version	✔	✘
	name	✔	✘
	port	✔	✘
	portlist	✔	✘
	protocol	✔	✘
SNMPService (3.3.23)	command	✘	✘
	contextEngineID	✔	✘
	contextName	✔	✘
	messageProcessingModel	✔	✘
	oid	✔	✘
	securityLevel	✔	✘
	securityModel	✔	✘
	securityName	✔	✘
Source (3.3.8)	ident	✔	✔
	interface	✔	✘
	Node	✔	✔
	Process	✔	✔
	Service	✔	✔
	spoofed	✔	✔
	User	✔	✔
Target (3.3.9)	decoy	✔	✔
	File	✔	✔
	ident	✔	✔
	interface	✔	✘
	Node	✔	✔
	Process	✔	✔
	Service	✔	✔
	User	✔	✔
ToolAlert (3.3.2)	alertident	✔	✔
	command	✔	✔
	name	✔	✔
User (3.3.18)	category	✔	✔
	ident	✔	✔
	UserId	✔	✔

Klasse	Attribut oder Unterklasse	D	LS
UserId (3.3.19)	ident	✔	✔
	name	✘	✘
	number	✘	✘
	tty	✔	✘
	type	✔	✔
WebService (3.3.22)	arg	✘	✘
	cgi	✔	✘
	http-method	✔	✔
	url	✔	✘

3.3 IDMEF

IDMEF (The Intrusion Detection Message Exchange Format) ist ein im RFC 4765 beschriebenes XML-Format aus dem Jahr 2007. Es besteht im Wesentlichen aus den beiden Klassen Alert und Heartbeat und deren Unterklassen.

3.3.1 Alert

Die Alert-Klasse enthält alle notwendigen Informationen, um einen Angriff oder Angriffsversuch vollständig beschreiben zu können. Die Mehrzahl aller Nachrichten, die im Projekt GIDS versendet werden, sind dieser Klasse zuzuordnen.

Unterklassen

Name	Erklärung	Anzahl
AdditionalData	siehe 3.3.11	0 bis ∞
Analyzer	siehe 3.3.6	1
Assessment	siehe 3.3.10	0 oder 1
AnalyzerTime	Die momentane Zeit des Analyzers.	0 oder 1
Classification	siehe 3.3.7	1
CreateTime	Ein Zeitstempel, der angibt, wann die Alarmmeldung erstellt wurde.	1
DetectTime	Ein Zeitstempel, der angibt, wann (zum ersten Mal) eine verdächtige Aktion bemerkt wurde. Das Erstellen der Meldung (CreateTime) kann auch später erfolgen.	0 oder 1
Source	siehe 3.3.8	0 bis ∞
Target	siehe 3.3.9	0 bis ∞

Spezialisierungen

Name	Erklärung	Anzahl
CorrelationAlert	siehe 3.3.3	0 oder 1
OverflowAlert	siehe 3.3.4	0 oder 1
ToolAlert	siehe 3.3.2	0 oder 1

Attribute

Name	Erklärung	Anzahl
messageid	Ein eindeutiger Identifier für die Alarmmeldung.	0 oder 1

3.3.2 ToolAlert

Wurde für den Angriff ein Tool verwendet und dieses Tool wurde durch die Sensorik erkannt, so können spezielle Informationen über dieses Programm in der Klasse ToolAlert übermittelt werden.

Unterklassen

Name	Erklärung	Anzahl
alertident	Eine Liste von Alarm-Identifiern (siehe Attribute der Alert-Klasse in 3.3.1). Da domänenübergreifend nicht garantiert werden kann, dass die Identifier eindeutig sind, kann die Angabe eines jeden Alarm-Identifiers durch die Angabe des Analyzer-Identifiers (siehe Attribute der Analyzer-Klasse in 3.3.6) ergänzt werden.	1 bis ∞
command	Das Kommando oder die Operation, die das Tool ausführen sollte, beispielsweise `BackOrifice ping`.	0 oder 1
name	Ein Bezeichner für den Alarm. Kann beispielsweise der Name des verwendeten Tools sein.	1

3.3.3 CorrelationAlert

Mit Hilfe der CorrelationAlert-Klasse können zusammengehörige Alarmmeldungen markiert werden.

Unterklassen

Name	Erklärung	Anzahl
alertident	Eine Liste von Alarm-Identifiern (siehe Attribute der Alert-Klasse in 3.3.1). Da domänenübergreifend nicht garantiert werden kann, dass die Identifier eindeutig sind, kann die Angabe eines jeden Alarm-Identifiers durch die Angabe des Analyzer-Identifiers (siehe Attribute der Analyzer-Klasse in 3.3.6) ergänzt werden.	1 bis ∞
name	Ein Bezeichner für den Korrelationsalarm, beispielsweise *massiver SSH-Scan.*	1

3.3.4 OverflowAlert

Speziell für Buffer Overflows existiert eine eigene Alarmklasse.

Unterklassen

Name	Erklärung	Anzahl
(ls)(d)buffer	Der Inhalt des Buffers, soweit er vom Sensor ermittelt werden konnte.	0 bis 1
program	Das Programm, das den Angriff auszuführen versucht. (Also nicht das angegriffene Programm)	1
size	Die Größe des Buffers	0 bis 1

Erläuterung

buffer. Der Buffers kann jeglichen Inhalt enthalten, da es hier stark vom Programm abhängt, was im Buffer steht. Es ist somit a priori der Inhalt dieses Feldes nicht abschätzbar.

3.3.5 Heartbeat

Jeder Sensor schickt in regelmäßigen Abständen Heartbeat-Meldungen an die zentralen Einheiten, um zu zeigen, dass er noch funktionsfähig ist. Zwar werden am Ende nicht alle Heartbeats aller beteiligten Sensoren an alle anderen Sites geschickt, innerhalb der GIDS-Infrastruktur müssen aber trotzdem Heartbeats verschickt werden, um zu zeigen, welche teilnehmenden Sites noch erreichbar sind.

Unterklassen

Name	Erklärung	Anzahl
AdditionalData	siehe 3.3.11	0 bis ∞
Analyzer	siehe 3.3.6	1
AnalyzerTime	Die momentane Zeit des Analyzers.	0 bis 1
CreateTime	Ein Zeitstempel, der angibt, wann die Heartbeatmeldung erstellt wurde.	1
Heartbeat-Intervall	Das Intervall, nach dem eine neue Heartbeatnachricht generiert wird.	0 bis 1

Attribute

Name	Erklärung	Anzahl
messageid	Ein eindeutiger Identifier für die Heartbeatmeldung.	0 bis 1

3.3.6 Analyzer

Die Informationen über den oder die Sensoren, die für die Erkennung und Weiterleitung der Alarm- oder Heartbeatmeldungen verantwortlich sind, werden in der Analyzer-Klasse zusammengefasst. Sind mehrere Sensoren an der Erkennung oder Weiterleitung beteiligt, werden diese in absteigender Reihenfolge rekursiv im XML-Baum repräsentiert, das heißt, das kleinste Kindelement hat als erstes eine verdächtige Aktion registriert und die Alarmmeldung generiert. Alle anderen Sensoren haben dann nur noch Informationen ergänzt, geändert oder die Meldung weitergeleitet.

Unterklassen

Name	Erklärung	Anzahl
Analyzer	siehe 3.3.6	0 bis 1
Node	siehe 3.3.16	0 bis 1
Process	siehe 3.3.20	0 bis 1

Attribute

Name	Erklärung	Anzahl
analyzerid	Ein eindeutiger Identifier für den Sensor. Zwar ist die Angabe einer analyzerid strenggenommen optional, sie wird jedoch verpflichtend, wenn in irgendeinem anderen Kontext ein Identifier gesetzt wird, zum Beispiel eine messageid der Heartbeat-Klasse.	0 bis 1
class	Der Typ des Sensors	0 bis 1
(ls)manufacturer	Der Hersteller des Sensors	0 bis 1
(ls)model	Die genaue Modellbeschreibung des Sensors	0 bis 1
(ls)name	Ein frei wählbarer Name, der die Identifizierung des Sensors für Menschen erleichtert.	0 bis 1
(ls)ostype	Der Name des Betriebssystems. Ist im Allgemeinen die Ausgabe des „uname s“-Kommandos.	0 bis 1
(ls)osversion	Die Version des Betriebssystems. Ist im Allgemeinen die Ausgabe des „uname r“-Kommandos.	0 bis 1
(ls)version	Die Versionsnummer	0 bis 1

Erläuterung

Sensoren gehören zu den kritischen Infrastrukturen und die gemachten Angaben könnten einem potentiellen Angreifer Interna preisgeben, die eventuell für Angriffe ausgenutzt werden könnten, vor allem wenn die Sensoren auf Systemen installiert sind, die nicht immer aktuell gehalten werden (können).

manufacturer, model & version. Ist bekannt, welche Sensoriken verwendet wird, kann ein Angreifer versuchen, unter dem Radar dieser Produkte zu bleiben.

name. Wird der Wert nicht durch einen neuen Namen ersetzt, ist in der Standardinstallation ein Rückschluss auf den verwendeten Sensor möglich.

ostype & osversion. Da jedes Betriebssystem eigene Sicherheitslücken hat, kann diese Angabe einem Angreifer helfen, die Schutzmaßnahmen zu umgehen.

3.3.7 Classification

Die Classification-Klasse gibt einem Alarm einen „Namen“. Sie hilft, Meldungen zu Alarmklassen zuzuordnen.

Unterklassen

Name	Erklärung	Anzahl
Reference	siehe 3.3.15	0 bis ∞

Attribute

Name	Erklärung	Anzahl
ident	Ein Identifier für diese Alarmklassifikation.	0 bis 1
text	Ein Text, der den Alarm klassifiziert, beispielsweise „PortScan“. Anmerkung: In diesem Attribut wird der Name der Klassifizierung von Alarmen angegeben. Dieser enthält keine personenbezogenen Informationen.	1

3.3.8 Source

Wurde von einem Sensor eine oder mehrere eindeutige Quellen von verdächtigen Aktionen identifiziert, so werden die darüber verfügbaren Informationen in der Source-Klasse abgelegt.

Unterklassen

Name	Erklärung	Anzahl
Node	siehe 3.3.16	0 bis 1
Process	siehe 3.3.20	0 bis 1
Service	siehe 3.3.21	0 bis 1
User	siehe 3.3.18	0 bis 1

Attribute

Name	Erklärung	Anzahl
ident	Ein eindeutiger Identifier für diese Instanz der Source-Klasse.	0 bis 1
(ls)interface	Sind mehrere Netzwerkinterfaces an dem angegriffenen Rechner vorhanden, kann hier das betroffene Interface genannt werden.	0 bis 1
spoofed	Wenn vom Sensor erkannt werden kann, dass die IP-Adresse des Angreifers gefälscht wurde, kann dies in diesem Attribut vermerkt werden.	0 bis 1

Erläuterung

interface. Die Angabe der intern genutzten Interfaces kann eventuell nicht erwünscht sein.

3.3.9 Target

Analog zur Source-Klasse stellt die Target-Klasse Informationen zu dem Ziel oder den Zielen einer verdächtigen Aktion bereit.

Unterklassen

Name	Erklärung	Anzahl
File	siehe 3.3.24	0 bis ∞
Node	siehe 3.3.16	0 bis 1
Process	siehe 3.3.20	0 bis 1
Service	siehe 3.3.21	0 bis 1
User	siehe 3.3.18	0 bis 1

Attribute

Name	Erklärung	Anzahl
decoy	Wenn vom Sensor erkannt werden kann, dass die IP-Adresse des Zielrechners gefälscht wurde, kann dies in diesem Attribut vermerkt werden.	0 bis 1
ident	Ein eindeutiger Identifier für diese Instanz der Target-Klasse.	0 bis 1
(ls)interface	Sind mehrere Netzwerkinterfaces an dem angegriffenen Rechner vorhanden, kann hier das betroffene Interface genannt werden.	0 bis 1

Erläuterung

interface. Die Angabe der intern genutzten Interfaces kann eventuell nicht erwünscht sein.

3.3.10 Assessment

Die Assessment-Klasse bietet den Sensoren die Möglichkeit, eine erste Einschätzung der verdächtigen Aktion zu machen. Weiterhin können die ergriffenen Gegenmaßnahmen protokolliert werden.

Unterklassen

Name	Erklärung	Anzahl
Action	siehe 3.3.13	0 bis ∞
Confidence	siehe 3.3.14	0 bis 1
Impact	siehe 3.3.12	0 bis 1

3.3.11 AdditionalData

Alle Daten, für die es keinen speziellen Platz in der IDMEF-Nachricht gibt, beispielsweise die Header-Daten eines IP-Paketes, können nach ihrem Datentyp sortiert in die AdditionalData mit hinzugefügt werden.

Attribute

Name	Erklärung	Anzahl
meaning	Eine kurze Beschreibung der in den AdditionalData versendeten Informationen.	0 bis 1
type	Ein Wert aus dem Wertbereich `boolean`/ `byte`/ `character`/ `date-time`/ `integer`/ `ntpstamp`/ `portlist`/ `real`/ `string`/ `byte-string`/ `xmltext`	0 bis 1

Inhalt

(ls)(d)Der Inhalt von AdditionalData kann dazu verwendet werden, um weitere Daten, die sonst im IDMEF-Standard keinen Platz gefunden haben, zu übermitteln oder um den IDMEF-Standard beliebig zu erweitern.

Erläuterung

Inhalt. Das Feld AdditionalData ist im IDMEF-Standarddazu gedacht, weitere Angaben, die für die Analyse eines Angriffs wichtig sind, mit angeben zu können oder den IDMEF-Standard an dieser Stelle zu erweitern. Was genau in den einzelnen Feldern gespeichert wird, kann man a priori aber nicht sagen.

3.3.12 Impact

Um eine Einschätzung der Auswirkungen einer verdächtigen Aktion geben zu können, ist die Impact-Klasse vorhanden.

Attribute

Name	Erklärung	Anzahl
completion	Ist ein Angriff erfolgreich gewesen? Kann der Sensor diese Frage beantworten, kann er sein Ergebnis in dieses Attribut schreiben.	0 bis 1
severity	Eine Einschätzung der Schwere der Auswirkung von Info bis hoch.	0 bis 1
type	Eine Kategorisierung der verdächtigen Aktion in verschiedene Typen ist mit Hilfe dieses Attributs möglich.	0 bis 1

Inhalt

(ls)(d) Textuelle Beschreibung der Auswirkung, soweit das vom Sensor unterstützt wird.

Erläuterung

Inhalt. Bei der Beschreibung der Auswirkung kann es unter Umständen passieren, dass personenbezogene Daten oder Daten, die unter dem Schutz der Sicherheitsrichtlinie stehen, weitergegeben werden. Beispielsweise könnte in der Meldung „Benutzerkennung XYZ kompromittiert“ stehen

3.3.13 Action

Wurde durch den Sensor irgendeine Aktion getriggert, beispielsweise eine Umkonfiguration der Firewall, so können diese Aktionen hier protokolliert werden.

Attribute

Name	Erklärung	Anzahl
category	Der Typ der durchgeführten Aktion.	1

Inhalt

(ls)(d)Erweiterte Beschreibung der Aktion.

Erläuterung

Inhalt. Der Inhalt kann personenbezogene Daten enthalten, wenn die Aktion auf einen speziellen Benutzer ausgerichtet ist, beispielsweise „blocked user XY“. Weiterhin kann es die lokale Sicherheitsrichtlinie verletzen, wenn der Name oder die Position der Firewall genannt wird.

3.3.14 Confidence

Je nachdem, aus welchen Regelsätzen eine Meldung generiert wurde, kann der Sensor eine Einschätzung geben, in wie weit er sich bei den gemachten Angaben „sicher“ ist und diese in der Confidence-Klasse ablegen. So ist es bei einer neuartigen heuristischen Meldung eher wahrscheinlich, dass ein gemeldeter Alarm sich am Ende als falsch erweist, als bei einer Signatur, die schon länger im Einsatz ist und nie Falschalarme generiert hat.

Attribute

Name	Erklärung	Anzahl
rating	Eine Einschätzung, die die eigene Aussagekraft in der Skala von „wenig“ zu „hoch“ oder in einem beliebigen vom Sensor gelieferten numerischen Wert bewertet.	1

Inhalt

Genauere Beschreibung der Einschätzung, beispielsweise ein genauer numerischer Wert zwischen 0.0 und 1.0.

3.3.15 Reference

Da Alarme von unterschiedlichen Herstellern von Sicherheitssystemen unterschiedlich genannt werden, ist eine einfache Korrelation nicht ohne weiteres gegeben. Daher bietet die Reference-Klasse Beschreibungen unterschiedlicher Hersteller zu verknüpfen und somit eine Möglichkeit bereit zu stellen, Korrelationen zu ermöglichen.

Unterklassen

Name	Erklärung	Anzahl
name	Der Name des Alarms. Stammt aus der in orgin genannten Quelle.	1
url	Ein Querverweis, auf dem weitere Informationen über den Alarm gefunden werden können.	1

Attribute

Name	Erklärung	Anzahl
meaning	Dieses Feld bietet die Möglichkeit, eine eigene Interpretation der Alarmmeldung zu geben.	0 bis 1
orgin	Der Quelle, aus der der Name des Alarms (name) stammt.	1

3.3.16 Node

Hosts und andere Netzwerkgeräte können durch die Node-Klasse identifiziert werden.

Unterklassen

Name	Erklärung	Anzahl
Address	siehe 3.3.17	0 bis ∞
(ls)location	Der Ort des Gerätes.	0 bis 1
(ls)(d)name	Der Name des Gerätes.	0 bis 1

Attribute

Name	Erklärung	Anzahl
category	Die Domäne, in der sich das Gerät befindet, beispielsweise „AFS“ oder „Windows NT“	0 bis 1
ident	Ein eindeutiger Identifier für diese Instanz der Node-Klasse.	0 bis 1

Erläuterung

location. Die genaue Angabe des Standortes eines Knotens kann aus sicherheitstechnischen Gesichtspunkten unerwünscht sein.

name. Der Namen kann personenbezogene Daten enthalten. Weiterhin kann diese Angabe aus sicherheitstechnischen Gesichtspunkten unerwünscht sein.

3.3.17 Address

Um einen Nutzer oder ein Gerät identifizieren zu können, gibt es in IDMEF die Möglichkeit, Adressinformationen in der Address-Klasse mit abzubilden. Diese reichen von IP-Adressen über MAC-Adressen bis hin zu E-Mail-Adressen.

Unterklassen

Name	Erklärung	Anzahl
(ls)(d)address	Die Adressinformation, beispielsweise die IP-Adresse.	1
(ls)netmask	Die Subnetzmaske der IP-Adresse.	0 bis 1

Attribute

Name	Erklärung	Anzahl
category	Die Kategorie, in der die Adressinformation (address) angegeben ist, beispielsweise IPv4-Adresse oder IPv6-Adresse.	0 bis 1
ident	Ein eindeutiger Identifier für die Instanz der Adress-Klasse.	0 bis 1
(ls)vlan-name	Der Name des VLANs, zu dem die Adresse gehört.	0 bis 1
(ls)vlan-num	Die Nummer des VLANs, zu dem die Adresse gehört.	0 bis 1

Erläuterung

Die Angabe einer Adresse ist sowohl für die Quelle und das Ziel eines Angriffs von entscheidender Bedeutung. Ohne die Angabe einer Netzwerkadresse ist beispielsweise ein DoS-Angriff, der von einer einzelnen Adresse ausgeht nicht zu erkennen. Auf der anderen Seite stellen das Datenschutzgesetz (BDSG) und die heutige Rechtsprechung eindeutig klar, dass beispielsweise IP-Adressen als personenbezogene Daten einem besonderen Schutz unterliegen und nicht ohne weiteres anlassunabhängig weitergegeben werden dürfen.

address. Die IP-Adresse, aber auch eine E-Mail-Adresse, ist klar ein personenbezogenes Datum und darf deshalb ohne besonderen Grund nicht weitergegeben werden. Ebenso kann es aus Sicht der lokalen Sicherheitsrichtlinie unerwünscht sein, beispielsweise interne Adressen nach außen zu geben.

netmask, vlan-name & vlan-num. Durch diese Angaben gibt man Interna preis, wie das Netz innen aufgebaut ist. Diese Angabe kann aus sicherheitstechnischen Gesichtspunkten unerwünscht sein.

3.3.18 User

In der User-Klasse kann ein Benutzer beschrieben werden. Sie dient jedoch hauptsächlich als Container für die UserId-Klasse.

Unterklassen

Name	Erklärung	Anzahl
UserId	siehe 3.3.19	1 bis ∞

Attribute

Name	Erklärung	Anzahl
category	Der Typ, der repräsentiert wird. Es kann sich dabei um einen Application-, ein Betriebssystembenutzer oder einen unbekannten Status handeln.	0 bis 1
ident	Ein eindeutiger Identifier für die Instanz der Benutzer-Klasse.	0 bis 1

3.3.19 UserId

In der UserId-Klasse können die Angaben zu einem Benutzer spezifiziert werden.

Unterklassen

Name	Erklärung	Anzahl
(ls)(d)name	Ein Benutzer- oder Gruppenname.	0 bis 1
(ls)(d)number	Eine Benutzer- oder Gruppennummer.	0 bis 1

Attribute

Name	Erklärung	Anzahl
ident	Ein eindeutiger Identifier für die Instanz der UserId-Klasse.	0 bis 1
(ls)tty	Das momentan vom Benutzer verwendete TTY.	0 bis 1
type	Der Typ der Userkennung. Wurde beispielsweise ein sudo-Befehl ausgeführt, ist es wichtig zu wissen, ob die angegebene Userkennung diejenige ist, die privilegiert ist oder nicht.	0 bis 1

Erläuterung

name. Zwar ist die Benutzerkennung „root“ datenschutzrechtlich nicht bedenklich, bei personalisierten Kennungen ist diese jedoch wenigstens innerhalb einer Domäne zumeist eineindeutig einer bestimmten Person zuzuordnen. Daher fällt diese Angabe wie auch die Angabe über die IP-Adresse unter das BDSG.

number. Innerhalb eines Systems ist eine Benutzernummer immer eindeutig einem Benutzer zuzuordnen. Somit ist die Angabe datenschutzrechtlich interessant.

tty. Diese Angabe erlaubt Rückschlüsse auf installierte Software und ist deshalb sicherheitstechnisch von Bedeutung.

3.3.20 Process

Für die Erkennung von Angriffen interessante Prozesse können in der Process-Klasse erläutert werden.

Unterklassen

Name	Erklärung	Anzahl
(ls)(d)arg	Wurden zum Ausführen des Programms Argumente mit übergeben, so werden diese hier in genau der Reihenfolge ihrer Angabe mit eingefügt.	0 bis ∞
(ls)(d)env	Benutzt, das Programm Umgebungsvariablen, so können diese hier mit angegeben werden.	0 bis ∞
(ls)name	Der Name des momentan ausgeführten Programms ohne Pfad- und Argumentangaben.	1
(ls)(d)path	Der vollständige Pfad zum Programm.	0 bis 1
pid	Die Prozess-ID des Programms.	0 bis 1

Attribute

Name	Erklärung	Anzahl
ident	Ein eindeutiger Identifier für die Instanz der Prozess-Klasse.	0 bis 1

Erläuterung

arg & env. In den übergebenen Argumenten ähnlich wie in den Umgebungsvariablen können sich sowohl bedenkliche Daten als auch datenschutzrechtlich relevante Daten befinden, beispielsweise bei den Argumenten des Befehls „ssh root@lrz.de“ ist eine Benutzerkennung mit angegeben worden.

name. Bei dem gemeldeten Prozess kann es sich um einen aus Sicht der Sicherheitsrichtlinie kritischen Prozess handeln, dessen Name eventuell nicht weitergegeben werden sollte.

path. Da bei der Angabe des Pfades unter Umständen der Benutzername auslesbar ist, wie beispielsweise in Home-Verzeichnissen, ist dieses Feld datenschutzrechtlich von Bedeutung.

3.3.21 Service

In der Service-Klasse können Netzwerk-Services auf Quell- oder Zielrechnern beschrieben werden.

Unterklassen

Name	Erklärung	Anzahl
(ls)name	Der Name des Services. Wenn möglich sollte der IANA-Name von bekannten Ports verwendet werden.	0 bis 1
(ls)port	Die verwendete Portnummer.	0 bis 1
(ls)portlist	Sind mehrere Ports in Verwendung, so kann man hier eine Liste aller Ports angeben.	0 bis 1
(ls)protocol	Ergänzende Angaben über das verwendete Protokoll können hier niedergeschrieben werden.	0 bis 1

Attribute

Name	Erklärung	Anzahl
(ls)iana_protocol_name	Der IANA-Protokollname.	0 bis 1
(ls)iana_protocol_number	Die IANA-Protokollnummer.	0 bis 1
ident	Ein eindeutiger Identifier für die Instanz der Service-Klasse.	0 bis 1
(ls)ip_version	Die IP-Version.	0 bis 1

Spezialisierung

Name	Erklärung	Anzahl
SNMPService	siehe 3.3.23	0 bis 1
WebService	siehe 3.3.22	0 bis 1

Erläuterung

iana_protocol_name, iana_protocol_number, ip_version, name, port, portlist & protocol. Diese Angaben geben viele Informationen weiter, wobei dies aus sicherheitstechnischen Überlegungen unter Umständen nicht erwünscht ist.

3.3.22 WebService

Als Spezialisierung der Service-Klasse bietet die WebService-Klasse die Möglichkeit ergänzende Angaben zu Web-Anwendungen anzugeben.

Unterklassen

Name	Erklärung	Anzahl
(ls)(d)arg	Die Argumente, die dem CGI-Skript mit übergeben wurden.	0 bis 1
(ls)cgi	Wurde ein CGI-Skript angefordert, so kann dieses hier benannt werden. Argumente, die dem CGI-Skript mit übergeben werden, werden aber in arg angegeben.	0 bis 1
http-method	Die http-Methode (POST oder GET).	0 bis 1
(ls)url	Die URL des Aufrufs.	1

Erläuterung

arg. In den übergebenen Argumenten können sich sowohl bedenkliche Daten als auch datenschutzrechtlich relevante Daten befinden, beispielsweise wenn bei den Argumenten eine Benutzerkennung mit angegeben wurde.

cgi & url. Aus Sicht der lokalen Sicherheitsrichtlinie ist die Weitergabe dieser Informationen eventuell zu beschränken, um Angriffsvektoren zu verkleinern.

3.3.23 SNMPService

Speziell für SNMP-Verkehr wurde die SNMPService-Klasse als eine eigene Spezialisierung der Service-Klasse mit eingeführt. Die anzugebenen Attribute beziehen sich zumeist auf RFC 3411.

Unterklassen

Name	Erklärung	Anzahl
(ls)(d)command	Das an den SNMP-Server verschickte Kommando.	0 bis 1
(ls)context-EngineID	Der Kontext-Engine-Identifier des Objektes.	0 bis 1
(ls)contextName	Der Kontextname des Objektes.	0 bis 1
(ls)message-Processing-Model	Die SNMP-Version.	0 bis 1
(ls)oid	Der Objekt-Identifier des Requests.	0 bis 1
(ls)securityLevel	Das Security-Level des Requestes.	0 bis 1
(ls)securityModel	Angabe, welches Security-Modell benutzt wurde.	0 bis 1
(ls)securityName	Der Security-Name des Objektes.	0 bis 1

Erläuterung

command, contextEngineID, contextName, messageProcessingModel, oid, securityLevel, securityModel & securityName. Diese Angaben geben viele Informationen weiter, wobei dies aus sicherheitstechnischen Überlegungen unter Umständen nicht erwünscht ist.

3.3.24 File

Alle Dateizugriffe, die im Zusammenhang mit einer verdächtigen Aktion stehen, können in der File-Klasse protokolliert werden.

Unterklassen

Name	Erklärung	Anzahl
access-time	Der Zeitpunkt des letzten Zugriffs auf die Datei.	0 bis 1
Checksum	siehe 3.3.28	0 bis ∞
create-time	Der Erstellungszeitpunkt der Datei.	0 bis 1
data-size	Die Größe der Datei in Bytes.	0 bis 1
(ls)disk-size	Die tatsächlich belegte Größe auf dem Datenträger in Bytes.	0 bis 1
FileAccess	siehe 3.3.25	0 bis ∞
Inode	siehe 3.3.27	0 bis 1
Linkage	siehe 3.3.26	0 bis ∞
modify-time	Der Zeitpunkt der letzten Änderung an der Datei.	0 bis 1
(ls)(d)name	Der Dateiname.	1
(ls)(d)path	Der Pfad zur Datei.	1

Attribute

Name	Erklärung	Anzahl
category	Der Kontext der Informationen, also ob die Informationen sich auf den Zeitpunkt vor der verdächtigen Aktion beziehen oder auf einen späteren Zeitpunkt.	1
(ls)file-type	Der MIME-Typ der Datei.	0 bis 1
(ls)fstype	Der Typ des Filesystems, beispielsweise FAT oder NTFS.	0 bis 1
ident	Ein eindeutiger Identifier für die Instanz der File-Klasse.	0 bis 1

Erläuterung

disk-size, file-type & fstype. Aus diesen Angaben kann man Rückschlüsse auf das verwendete System ziehen, die unter Umständen aus Sicht der lokalen Sicherheitsrichtlinie unerwünscht sind.

name. Bei der gemeldeten Datei kann es sich um einen aus Sicht der Sicherheitsrichtlinie kritischen Datei handeln, dessen Name eventuell nicht weitergegeben werden sollte oder der Dateiname erlaubt Rückschlüsse auf den Benutzer und ist daher aus datenschutzrechtlicher Sicht zu beanstanden.

path. Da bei der Angabe des Pfades unter Umständen der Benutzername auslesbar ist, wie beispielsweise in Home-Verzeichnissen, ist dieses Feld datenschutzrechtlich von Bedeutung.

3.3.25 FileAccess

Die FileAccess-Klasse repräsentiert die Zugriffsberechtigungen auf eine Datei.

Unterklassen

Name	Erklärung	Anzahl
(ls)Permission	Das Level der Zugriffsberechtigung von „kein Zugriff“ bis hin zu „Benutzer hat die volle Kontrolle über die Datei“.	1 bis ∞
UserId	siehe 3.3.19	1

Erläuterung

Permission. Die lokale Sicherheitsrichtlinie verbietet unter Umständen die Weitergabe der Angabe der genauen Berechtigungen. Somit soll ein möglicher Angriffsvektor verkleinert werden, indem offensichtliche Fehlkonfigurationen nicht öffentlich gemacht werden.

3.3.26 Linkage

In der Linkage-Klasse können Verknüpfungen im Dateisystem wie beispielsweise Hard-Links dargestellt werden.

Unterklassen

Name	Erklärung	Anzahl
File	siehe 3.3.24	1
(ls)(d)name	Der Dateiname.	1
(ls)(d)path	Der Pfad zur Datei.	1

Attribute

Name	Erklärung	Anzahl
category	Der Typ der Verknüpfung zwischen den Dateien, beispielsweise Hard-Links oder Shortcuts.	1

Erläuterung

name. Bei dem gemeldeten Prozess kann es sich um einen aus Sicht der Sicherheitsrichtlinie kritischen Prozess handeln, dessen Name eventuell nicht weitergegeben werden sollte.

path. Da bei der Angabe des Pfades unter Umständen der Benutzername auslesbar ist, wie beispielsweise in Home-Verzeichnissen, ist dieses Feld datenschutzrechtlich von Bedeutung.

3.3.27 Inode

Die Inode-Klasse wird verwendet, um zusätzliche Informationen, die in Inodes der Unix Dateisysteme enthalten sind, zu repräsentieren.

Unterklassen

Name	Erklärung	Anzahl
change-time	Der Zeitpunkt der letzten Änderung der Inode-Daten.	0 bis 1
c-major-device	Die Major-Device-Nummer der Datei, falls es sich um ein Character Special Device handelt.	0 bis 1
c-minor-device	Die Minor-Device-Nummer der Datei, falls es sich um ein Character Special Device handelt.	0 bis 1
major-device	Die Major-Device-Nummer des Geräts, auf dem die Datei sich befindet.	0 bis 1
minor-device	Die Minor-Device-Nummer des Geräts, auf dem die Datei sich befindet.	0 bis 1
number	Die Inode-Nummer.	0 bis 1

3.3.28 Checksum

Sind Checksummen von Dateien verfügbar und diese Information für die Analyse eines Vorfalls von Interesse, so können hier die benötigten Angaben gemacht werden.

Unterklassen

Name	Erklärung	Anzahl
(ls)key	Falls nötig, kann in diesem Attribut ein Schlüssel zur Berechnung der Checksum mit angegeben werden.	0 bis 1
value	Der berechnete Wert der Checksum.	1

Attribute

Name	Erklärung	Anzahl
algorithm	Der Krypto-Algorithmus, der zum Berechnen der Checksum verwendet wurde.	1

Erläuterung

key. Die Weitergabe eines kryptographischen Schlüssels, auch wenn er nur für die Berechnung von Checksummen verwendet wird, ist aus sicherheitstechnischen Überlegungen nicht anzuraten.

Kapitel 4

Betriebsmodell des GIDS-Dienstes

Im Rahmen des GIDS-Projektes ist ein Produktivsystem realisiert worden, um einen Dienst für den Betrieb eines föderierten IDS anbieten zu können. In diesem Kapitel wird ein vorbereitendes Betriebsmodell für einen GIDS-Dienst vorgestellt, das die Teilnahme und den Betrieb des GIDS-Dienstes umfasst. Allerdings muss berücksichtigt werden, dass im Rahmen der Umsetzung des Verwertungsplans das Betriebskonzept noch verfeinert wird und beispielsweise an kommerzielle Aspekte angepasst wird.

4.1 Verwendung der Daten

Der wichtigste Verwendungszweck ist die Erkennung von Angriffen im D-Grid. Diese erfolgt sowohl auf der Seite der beteiligten Partner als auch auf der Seite des Betreibers. Alle Daten, die von den Partner exportiert werden und damit die Filterung und Anonymisierung passieren, werden in die Datenbank des GIDS importiert. Dort werden sie vom Portal als korrelierte IDS-Meldungen und Statistiken dargestellt:

Auswertung durch die Partner Die beteiligten Partner haben im GIDS-Portal Zugriff auf die korrelierten IDS-Meldungen der eigenen Ressourcen. Ziel ist es, Angriffe anhand kritischer Meldungen und Anomalien in den Statistiken erkennen zu können. Auf der einen Seite kann der Partner nach Erkennung bzw. nach dem Verdacht eines Angriffes die Daten sichten, die in der lokalen Datenbank vorhanden sind. Dies ermöglicht die genaue Analyse des Angriffs. Auf der anderen Seite kann der Partner den Betreiber im Sinne der Frühwarnung alarmieren, der dann eine Korrelation im gesamten Datenbestand durchführt und eine Warnung an alle Teilnehmer versendet.

Auswertung durch den Betreiber Der GIDS-Betreiber hat die globale Sicht auf alle Daten und kann deshalb globale Anomalien erkennen. In diesem Fall werden alle Teilnehmer alarmiert, die dann die Ursache in den lokalen Datenbeständen analysieren können,

Des Weiteren werden die Daten optional an das Frühwarnsystem CarmentiS gesendet. Hier fließen diese in das Lagebild der Internets ein. Für die nach CarmentiS gesendeten Daten kann jeder Partner entscheiden, ob diese Daten für die Bearbeitung von Sicherheitsvorfällen genutzt werden sollen. Ist dies der Fall, werden diese zusätzlich von dem AW-Dienst des DFN-Vereins verarbeitet und an betroffene Einrichtungen im DFN gesendet.

Export der IDS-Sensordaten Jeder Partner kann in der administrativen Domäne frei wählen, welche Daten gefiltert und anonymisiert werden können. Weiterhin können Partner angeben, wann die Daten wieder gelöscht werden sollen. Dies geschieht auch beim Ausscheiden eines Partners.

Darstellung im GIDS-Portal Alle von den Partnern exportierten Daten werden im GIDS-Portal dargestellt. Der Zugriff ist für alle Teilnehmer am D-Grid offen, die über ein gültiges D-Grid-Zertifikat verfügen. Der Zugriff der Daten ist auf die Meldungen beschränkt, die jeweils dem angemeldeten Benutzer auf der Basis der Ressourcen zugeordnet werden können. Grundlage dafür bildet die GRRS-Datenbank des D-Grids.

4.2 Teilnahme am GIDS-Dienst

Die Teilnahme und Nutzung des GIDS-Dienstes umfasst die folgenden Punkte:

- Organisatorische und technische Maßnahmen zur technischen Anbindung der administrativen Domäne an das Grid-IDS. Auf organisatorischer Ebene ist dies die Registrierung der Partner beim GIDS-Betreiber. Des Weiteren findet auf technischer Ebene eine Anbindung an den GIDS-Bus zum Export von Daten statt.
- Eine Regelung für Partner zum Verlassen des GIDS-Dienstes.
- Der GIDS-Dienst beinhaltet die Anmeldung beim Portal und den Zugriff auf die Daten, auf die der angemeldete Benutzer berechtigten Zugriff hat.
- Den Erhalt von Meldungen durch das GIDS-Portal per E-Mail.
- Der Betreiber kann zusätzlich zum Betrieb die Rolle des Analysten übernehmen. Diese Rolle betrifft die Analyse der globalen Daten, die in Form des Lagebildes anfällt.

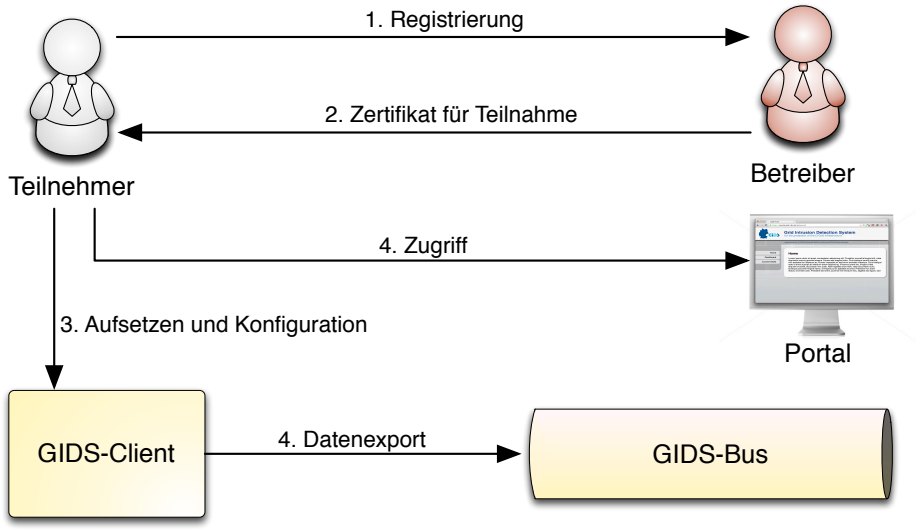


Abbildung 4.1: Überblick über die Teilnahme am GIDS-Dienst

Eine Übersicht über den Ablauf bei der Teilnahme am GIDS-Dienst ist in Abb. 4.1 gegeben und besteht aus den folgenden Schritten:

Registrierung Der erste Schritt ist die Registrierung beim Betreiber des GIDS. Die Registrierung dient dazu, Kontaktdaten zwischen dem Betreiber des GIDS-Dienstes und dem Partner auszutauschen.

Anbindung der administrativen Domäne an das GIDS Nach der Registrierung erfolgt der technische Anschluss der adminsitrativen Domäne des Partners an das GIDS. Dazu erhält dieser ein Zertifikat, dass den autorisierten Zugriff des GIDS-Client des Partners für den GIDS-Bus ermöglicht. Des Weiteren muss der Export konfiguriert werden. Dies

umfasst die Anonymisierung und Filterung der Sensor-Daten. Dazu wird vom GIDS-Projekt eine Konfiguration zur Verfügung gestellt, die alle Anforderungen an den Datenschutz und die zu erwartenden Sicherheitsanforderungen des Partners erfüllt. Mittels eines Test-Sensors lässt sich die Funktion des Exports und der Filterung der IDS-Meldungen überprüfen und eventuell vorhandene Fehler feststellen.

Betrieb des Sensorik Jeder Partner ist in der eigenen administrativen Domäne autonom. Dies bedeutet, dass sowohl die Sensorik als auch die Filterung und Anonymisierung unter der Hoheit der Partner stehen und frei gewählt werden können. Zwar wird empfohlen, die vom GIDS-Projekt unterstützten Sensoren Snort, Prelude-LML und OSSEC zu verwenden, allerdings lassen sich alle IDS-Sensoren einbinden, die die Prelude Bibliothek zum Datenexport unterstützen.

Zugriff auf das GIDS-Portal Nach dem Anschluss der administrativen Domäne, sind die exportierten Sensor-Daten im GIDS-Portal sichtbar. Der Zugriff auf das Portal erfolgt mit dem vorhandenen Grid-Zertifikat des Partners. Die Rechte sind entsprechend den Angaben in der GRRS-Datenbank des D-Grids gesetzt. Bei Problemen beim Zugriff sind also die Angaben in der GRRS-Datenbank zu prüfen und gegebenfalls zu korrigieren. Über den Zugang zum Portal lassen von dem Benutzer Abonnements für Meldungen anlegen, um über den aktuellen Zustand von Ressourcen per E-Mail informiert zu werden. Zusätzlich kann der Betreiber die Rolle des Analysten übernehmen und eine Analyse der globalen Daten durchführen, die in Form des Lagebildes anfällt.

Auscheiden aus dem GIDS-Dienst Jeder Partner kann zu einem beliebigem Zeitpunkt aus dem GIDS-Dienst ausscheiden. Dies beinhaltet, dass alle Daten des Partners aus dem GIDS gelöscht werden, soweit dies technisch durchführbar[1] ist.

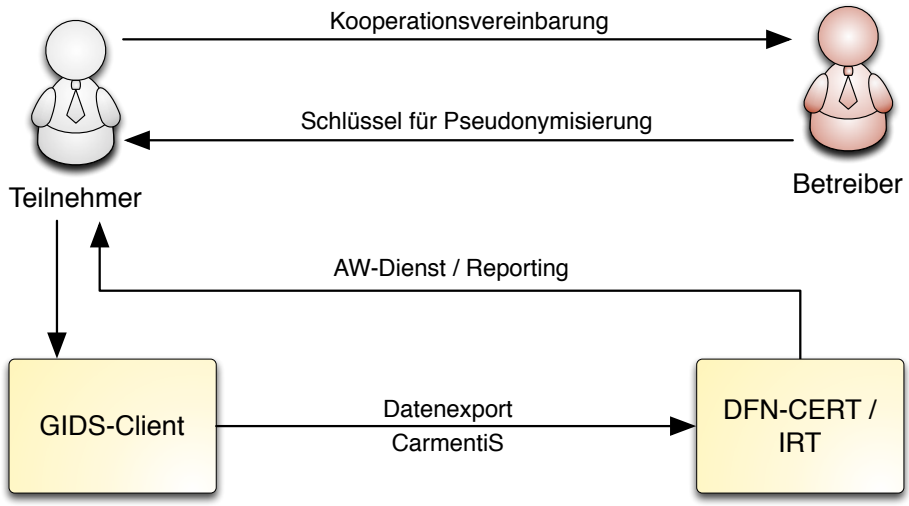


Abbildung 4.2: Überblick über die Teilnahme am GIDS-Dienst mit CarmentiS

4.3 Teilnahme am CarmentiS Datenexport und AW-Dienst

Ein erweitertes Betriebsmodell ist für die Anbindung und Kooperation mit CarmentiS vorhanden. CarmentiS ist ein IT-Frühwarnsystem, dass vom DFN-CERT betrieben wird. Eine Kooperation steht jedem GIDS-Partner auf freiwilliger Basis offen. Findet eine Kooperation statt, erhält der GIDS-Partner zusätzlich Zugriff auf das CarmentiS Lagebild. Neben dem

[1] Beispielsweise kann nicht garantiert werden, dass die Daten auf Back-ups gelöscht werden können

Export der IDS-Daten an GIDS werden die Daten zusätzlich an CarmentiS exportiert. Dort fließen diese in das Lagebild ein und können mit Zustimmung des Partners für die Bearbeitung von Sicherheitsvorfällen genutzt werden. Dazu betreibt der DFN-Verein den AW-Dienst, der Daten über kompromittierte Systeme sammelt und diese dann automatisiert den entsprechenden Einrichtungen für die Bearbeitung der Vorfälle zur Verfügung stellt. Das erweiterte Betriebsmodell betrifft die folgenden Punkte:

Registrierung bei CarmentiS Die Registrierung bei CarmentiS erfolgt über die Unterzeichnung der Kooperationsvereinbarung. Der Partner erhält dadurch Zugriff auf das CarmentiS Lagebild. Des Weiteren findet ein Austausch von Schlüsseln zur Pseudonymisierung der Daten statt.

Konfiguration des Datenexportes Der Datenexport basiert auf einem separaten Programm auf der Seite des Betreibers, der die Daten an CarmentiS weitergibt. Ein spezielle Konfiguration auf der Seite des Partners ist also nicht notwendig. Allerdings muss der Datenexport vom GIDS-Betreiber frei geschaltet werden. Weiterhin kann beim Export angegeben werden, dass die Daten auch für die Vorfallsbearbeitung verwendet werden können. Ein Vorteil ist, dass die Partner auf diesem Wege Warnmeldungen des AW-Dienstes für eigene Systeme erhalten können.

Teilnahme am AW-Dienst Die Teilnahme am AW-Dienst erfolgt über den DFN-Verein und ist für DFN-Anwender nutzbar. Die Konfiguration erfolgt über das DFN-Portal. Im Gegensatz zum GIDS-Portal ist der Zugriff auf einzelne Administratoren beschränkt und muss für diese separat frei geschaltet werden. Für jeden DFN-Anwender existiert allerdings mindestens ein Kontakt, der berechtigten Zugriff auf das Portal hat. Über diesen können Regeln für einzelne IP-Adressen oder Netzwerkblöcke konfiguriert werden. Im Rahmen des GIDS bietet sich der Einsatzzweck, Netzwerke zu konfigurieren, die für Grid-Systeme genutzt werden. Dies bietet die Möglichkeit, automatisch Warnmeldungen zu den Grid-Systemen innerhalb der eigenen administrativen Domäne zu erhalten.

Kapitel 5

Tragfähigkeitsnachweis des Gesamtsystems

Um die Tragfähigkeit des produktiven GIDS zu demonstrieren, werden in diesem Kapitel die bis zum Ende des Projektes erzielten Ergebnisse vorgestellt. Dabei dienen die Szenarien der Kompromittierung von Benutzer-Accounts und der Ausbreitung eines Internet-Wurms aus der Analyse der Bedrohungen in [9] als Motivation. Das erste Szenario ist insbesondere für Grid-Systeme von Bedeutung, weil ein großer Teil der beobachteten Vorfälle in Grids entweder aus der Kompromittierung von Benutzer-Accounts hervorging oder diese als Ziel hatte. Der zweite Punkt ist für die Bewertung eines Frühwarnsystems wichtig. Spezielles Ziel eines Frühwarnsystems ist die frühzeitige Erkennung von Internet-Würmern, um noch nicht betroffene Systeme noch schützen zu können. Hier spielt insbesondere die sinnvolle Zusammenarbeit der verteilten bzw. föderierten Systeme eine wichtige Rolle. Da während der Laufzeit des Projektes keine neuartigen Internet-Würmer auftraten, wurde die Ausbreitung bekannter Würmer simuliert. Ein weiterer Vorteil der Simulation ist deren Reproduzierbarkeit, die die Kalibrierung der Sensorik erleichtert.

5.1 Angriffserkennung durch die GIDS-Sensorik

5.1.1 Übersicht über erkannte Angriffe

Um die Effektivität der Sensorik zu testen, wurden mit Snort und Prelude-LML zwei unterschiedliche IDS im Rahmen des GIDS eingesetzt. In diesem Abschnitt werden die erkannten Angriffe und die daraus gewonnenen Erkenntnisse zusammengefasst.

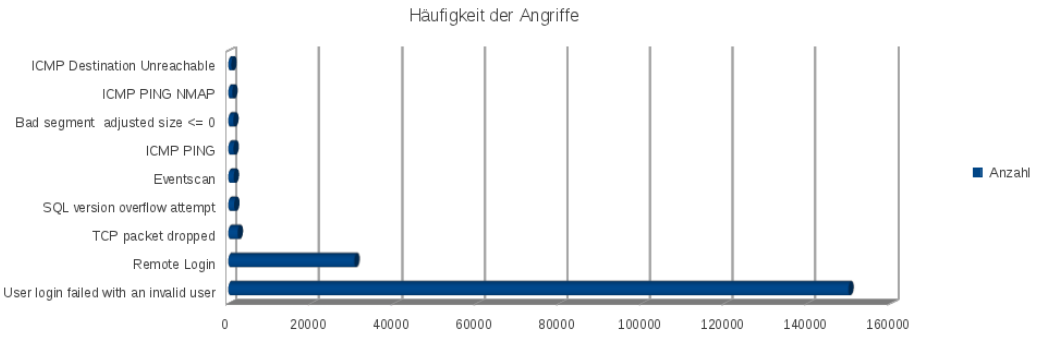


Abbildung 5.1: Häufigkeitsverteilung der erkannten Angriffe

Abb. 5.1 gibt eine Übersicht über die am häufigsten erkannten Angriffe. Die absoluten Zahlen sind in Tabelle 5.1 dargestellt. Der „Typ“ und die „Severity“ ergeben sich aus der Klassifizierung der Angriffe durch Prelude. Der Typ ist Teil des IDMEF-Formates und gibt das Ziel des Angriffs an. Dies kann beispielsweise sein, die Privilegien eines Benutzer (Typ:„user“) oder des Administrators (Typ:„admin“) zu übernehmen. Die Severity ist ebenfalls Teil des IDMEF-Klassifizierung und ist ein Maß für die Schwere des Angriffs. So ist es beispielsweise kritischer, wenn ein Angriff potentiell die Kompromittierung eines Systems ermöglicht, als wenn der Angriff nur zur Erkundung der Eigenschaften des Systems dient. Die Beschreibung wurde aus dem entsprechenden Feld der IDS-Regeln von Prelude übernommen.

Typ	**Severity**	**Anzahl**	**Beschreibung**
user	medium	149913	User login failed with an invalid user
admin	medium	30449	Remote Login
other	medium	2254	TCP packet dropped
other	high	1414	SQL version overflow attempt
other	high	1295	Eventscan
other	low	1248	ICMP PING
other	low	1197	Bad segment adjusted size <= 0
other	medium	954	ICMP PING NMAP
other	low	741	ICMP Destination Unreachable

Tabelle 5.1: Absolute Häufigkeitsverteilungen der IDS-Meldungen

Bei der Auswertung der Ergebnisse muss berücksichtigt werden, dass der Sensor auf einem Linux System betrieben wurde. Aus diesem Grund schlugen Angriffe der meisten Windows-Internet-Würmer ohne Verbindungsaufbau fehl und wurden nicht von dem Snort NIDS erkannt.

Bei den erkannten Angriffen haben sich Brute-Force-Angriffe auf schwache SSH-Passwörter als die Kategorie herausgestellt, die mit sehr großen Abstand am häufigsten aufgezeichnet wurde. Dies unterstreicht die Bedeutung dieser Angriffe für die Bedrohung von Grid Systemen. Der W32.Slammer-Wurm verbreitet sich durch Ausnutzung einer Schwachstelle im Microsoft SQL-Server (CVE-2002-0649). Im Gegensatz zu anderen Würmern findet der Angriff über UDP auf Port 1434 statt und es ist ein einzelnes UDP-Paket für die Weiterverbreitung ausreichend. Aus diesem Grund konnte der Wurm durch Snort erkannt werden („SQL version overflow attempt“). Die häufig beobachteten ICMP-Pings sind zwar in der Mehrzahl keine direkten Angriffe; jedoch wird dies häufig zur Vorbereitung für einen Angriff durchgeführt.

Abb. 5.2 zeigt die Häufigkeitsverteilung der Klassifizierung durch Prelude. Dabei wird zwischen den Klassen „high“, „medium“, „low“, und „info“ unterschieden, die jeweils eine Einschätzung der Bedrohung durch den Angriff wiedergeben. Insgesamt dominiert die Klasse „medium“, in der die „Brute-Force-Angriffe“ auf schwache SSH-Passwörter fallen. Die Kategorie „high“ wird im Wesentlichen durch Angriffe gebildet, die bei einem erfolgreichen Angriff eine direkte Kompromitierung des Systems ermöglichen. Hierunter fällt beispielsweise der Angriff des Slammer-Wurms. Die Klasse „low“ sind IDS-Meldungen, die zwar auf einen Angriff hindeuten können, allerding keine direkte Konsequenzen nach sich ziehen. Ein Beispiel ist die in auch in Abb. 5.1 gezeigte Meldung „TCP packet dropped“, die im Rahmen eines Denial-of-Service-Angriffes durch Erschöpfen der Bandbreite und dadurch verursachten Paketverlust auftreten kann. Allerdings kann auch ein Fehlalarm (False-Positive) die Ursache sein.

Aufgrund der Dominanz der Brute-Force-Angriffe auf schwache SSH-Passwörter, wird auf diese im Detail eingegangen.

Insgesamt wurden Angriffe aufgezeichnet, die von 170 unterschiedlichen IP-Adressen ausgingen. Die meisten Angriffe gingen, wie in Abb 5.3 dargestellt, von China, den USA und Deutschland aus. Dabei wurden von einer einzelnen IP im Maximum 67112 IDS-Meldungen erzeugt, was der gleichen Anzahl an Rateversuchen entspricht. Die Verteilung der Anzahl der Meldungen wird in Abb. 5.4 gezeigt. Dabei wird die Verteilungsfunktion näherungsweise durch eine exponentielle Abnahme der Anzahl der Versuche wiedergegeben.

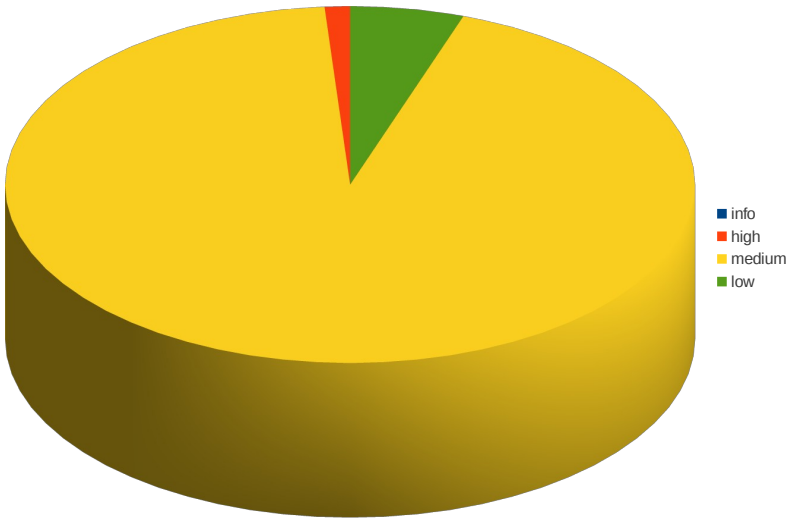


Abbildung 5.2: Häufigkeitsverteilung der Kategorien der erkannten Angriffe

Für eine Kompromittierung genügt ein erfolgreich geratenes oder ein bereits kompromittierter Passwort. Das bedeutet, dass die Anzahl der Versuche zwar einen Indikator für die Bedrohung darstellt, jedoch nicht als alleiniges Merkmal ausreicht. So werden beispielsweise bereits kompromittierte Passwörter oder Credentials auf anderen Systemen ausprobiert, die unter Umständen den kompromittierten Benutzer beheimaten. Als Resultat zeigt sich, dass eine Aggregation der Meldungen zu einem Vorfall sehr wichtig ist, die durch den Prelude-Correlator durchgeführt wird.

5.1.2 Auswertung der Eigenschaften der Korrelations-Alarme

Die Korrelation und Aggregation ist ein wichtiger Schritt bei der Auswertung der verteilten IDS-Meldungen in der föderierten IDS Umgebung des GIDS. Im Rahmen des GIDS-Projektes wird hierfür der Prelude-Correlator verwendet, der an die Anforderungen des Projektes angepasst wurde. Auf der einen Seite ist die Aufgabe, durch Zusammenfassen zusammengehöriger IDS-Meldungen bedrohliche Situationen besser zu erkennen. So können umfangreiche Brute-Force-Angriffe zu einem einzelnen Alarm zusammengefasst werden. Auf der anderen Seite ist eine wichtige Aufgabe, IDS-Meldungen zu unterdrücken, die entweder auf Fehlalarme hinweisen oder nur in einem erweiterten Zusammenhang auf Angriffe hindeuten. Um die Tragfähigkeit zu überprüfen, wurden die Korrelationsergebnisse ausgewertet, was zusammenfassend zu den folgenden Ergebnissen geführt hat:

- Aus den Log-Daten mehrerer Firewalls wurden IDS-Meldungen aus geblockten Verbindungen erzeugt. Zwar weist ein einzelner Alarm auf einen Angriff hin, jedoch wurde der Angriff bereits durch die Firewall abgewehrt und bedarf keiner weiteren Kenntnisnahme. Aufgrund des Einsatzes des Prelude-Correlators wurden diese Angriffe zusätzlich mit den Daten der DSHIELD-Datenbank korreliert, die vom SANS Internet Storm Center betrieben wird. In dieser werden bekannte Quellen von Angriffen gespeichert. Resultat der Korrelation ist, dass im GIDS-Portal Angriffe von bekannten, aggressiven Quellen dargestellt werden können.

- Wie geplant werden mehrere IDS-Meldungen zu Portscans oder Brute-Force-Angriffen zusammengefasst. Hierbei hat sich gezeigt, dass die ursprünglichen Regeln der Correlators in machen Fälle, fehlerhafter Weise mehrere IDS-Meldungen zu einem Portscan aggregiert hat. Grund hierfür waren fehlgeformte TCP-Pakete, die mehrfach IDS-

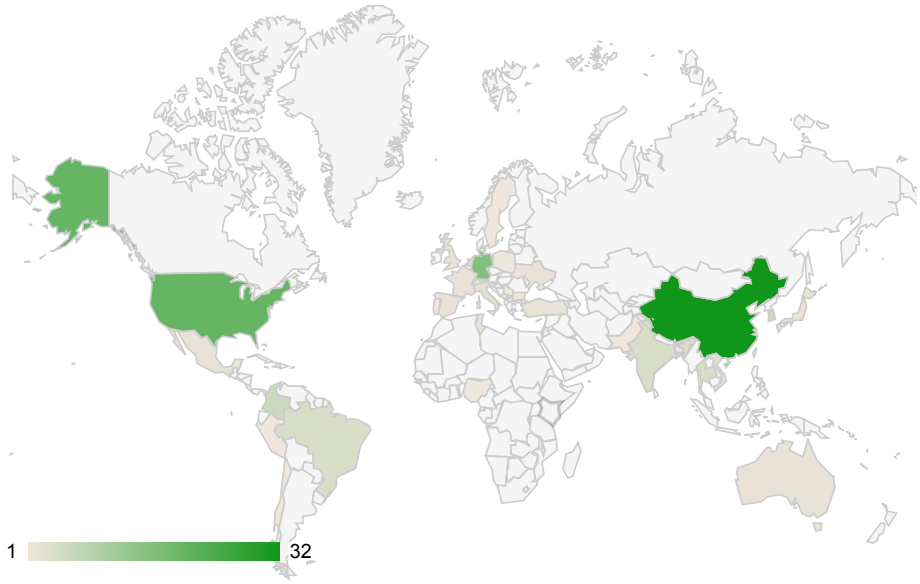


Abbildung 5.3: Ausgangspunkte von Brute-Force-Angriffen

Meldungen produzierten. Um diese Fehlalarme auszuschließen, wurde der Prelude-Correlator entsprechend angepasst.

- Für die Anonymisierung der IP-Adressen ist es ausreichend, das letzte Byte zu löschen oder durch eine Null zu ersetzen. Dies hat allerdings Auswirkungen auf die Korrelation der auf diese Weise anonymisierten IDS-Meldungen, weil dadurch die Trennung zwischen IP-Adressen im gleichen Class-C Netzblock wegfällt. Auf diese Problematik kann durch Hochsetzen von Schwellwerten im Prelude-Correlator reagiert werden.

- Um auf alle wichtigen IDS-Meldungen reagieren zu können, wird für jede Meldung ein korrelierter Alarm generiert, die von einem IDS als schwerwiegend klassifiziert (Severity: „high“) werden.

Zusammenfassend hat sich gezeigt, dass der Prelude-Correlator die gesetzten Aufgaben erfüllt. Allerdings ist es sinnvoll, den Prelude-Correlator im laufenden Betrieb kontinuierlich zu verbessern und an neue Bedrohungen anzupassen, um die Rate der Fehlalarme und die Korrelationsleistung weiter zu verbessern.

5.2 Detektion einer simulierten Wurmausbreitung

Im folgenden Abschnitt werden im zuvor vorgestellten Szenario und Aufbau des GIDS Daten einer simulierten Ausbreitung des Wurms *Code Red v2* und *Code Red II* beigefügt. Die induzierten Daten stellen dabei eine Art zusätzlichen virtuellen Log-Sensor einer Firewall der jeweiligen Domäne dar. Es werden zusätzlich zu den tatsächlich anfallenden Realdaten alle Access-List Hits einer zentralen Firewall der jeweiligen Domäne, die durch die Ausbreitung der Wurminstanzen verursacht worden wären, berichtet. Die simulierten Daten der Wurmausbreitung stammen dabei aus der von Harald Schmidt in [10] entwickelten Plattform zur Simulation verschiedener Wurmausbreitungsstrategien. Der Einfachheit halber beginnt die künstlich induzierte Wurmausbreitung stets um 0:00 Uhr.

Im weiteren Verlauf dieses Abschnitts werden zuerst Grundlagen zu den beiden Wurminstanzen *Code Red v2*, *Code Red II* und *FreeBSD.Scalper* gelegt. Die Erkennungsleistungen der einzelnen Koalitionspartner werden anschließend in Abschnitt 5.2.2 betrachtet, bevor der Betrieb des GIDS mit allen derzeitigen Koalitionspartnern in Abschnitt 5.2.3 beleuchtet wird.

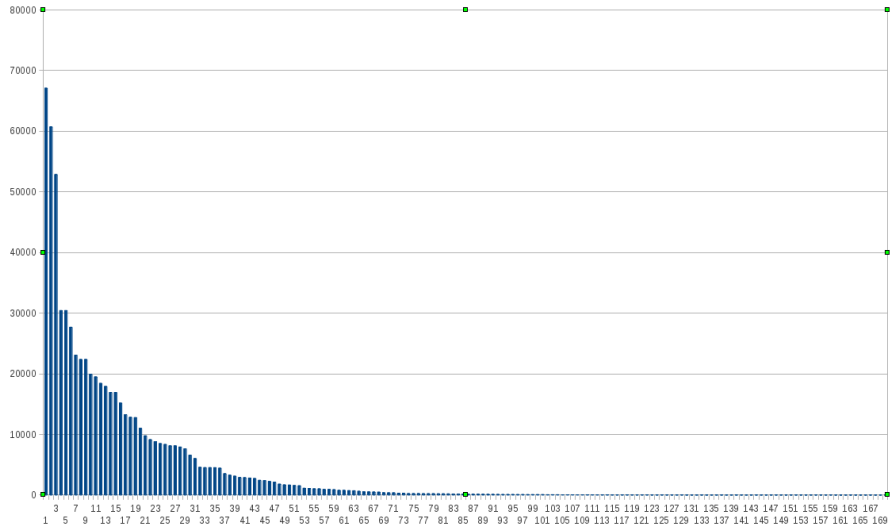


Abbildung 5.4: Häufigkeitsverteilung der Rateversuche der Brute-Force-Angriffe

Ein Vergleich der Effektivität und Effizienz der Erkennungsleistungen der einzelnen Kooperationspartner im Gegensatz zu einer kooperativen Erkennung aller Partner gemeinsam schließt diesen Abschnitt zusammenfassend ab.

5.2.1 Grundlagen: Computerwürmer und ihre Ausbreitung

Ein Wurm ist eine Schadsoftware, die sich nach ihrer Ausführung selbstständig auf andere Computer ausbreitet. Zu diesem Zwecke werden zumeist Sicherheitslücken eines Betriebssystems ausgenutzt. Würmer können Programmteile mit sich führen, die Schaden an den befallenen Computersystemen verüben. Typisch dabei ist zum Beispiel das Einrichten einer *Backdoor* auf dem befallenen System, so dass der vollständige Zugriff durch Dritte ermöglicht wird. Häufig ist auch der Versuch eines *Denial-Of-Service*-Angriffs (DoS) zu beobachten. Dabei wird versucht, die Betriebsfähigkeit des befallenen Computersystems so zu stören, dass das Computersystem seinen Dienst teilweise vorübergehend einstellt.

Nach der Definition der *Network Working Group* der *Internet Engineering Task Force* (IETF) in [1] propagiert sich ein Wurm über ein Netz zu einem anderen Computersystem. Dazu muss eine jede Wurminstanz nach neu verfügbaren Zielen suchen, die infizierbar sind. Bei der Suche werden auch viele gegen den entsprechenden Wurm immune Systeme erreicht, die nicht infiziert werden können. Dies bietet jedoch eine weitere Gefahr, da ein befallener Rechner durch seinen aktiven Suchvorgang offenbart, dass er von einem Wurm infiziert wurde und somit implizit eingesteht, über offene Sicherheitslücken oder geöffnete Hintertüren zu verfügen. Diese Information können sich weitere Angreifer zu nutze machen.

Eine typische Wurmausbreitung verläuft initial mit einem exponentiellen Wachstum bezüglich der Anzahl der infizierten Computersysteme. Zu Beginn breitet sich der Wurm nur sehr schleppend aus. Das liegt daran, dass noch sehr wenige befallene Rechner existieren, die weitere Computersysteme vereinnahmen können. Im Lauf der Zeit steigt die Anzahl der infizierten Rechner signifikant an und somit auch die Geschwindigkeit der Weiterverbreitung des Wurms. Es sind nun viele Systeme durch ihren Wurmbefall in der Lage, weitere Rechner zu infizieren. Nach einer Zeit setzt jedoch eine Sättigung ein, so dass die Anzahl der infizierten Systeme stagniert. Dies liegt zum einen daran, dass die Anzahl der noch nicht infizierten Rechner, die noch immer entsprechende Sicherheitslücken für einen potentiellen Befall bieten, im Laufe der Zeit sinkt. Zum anderen ist es vielen Systemadministratoren zwischenzeitlich möglich gewesen, entsprechende Patches einzuspielen und dadurch ihre Systeme gegen einen

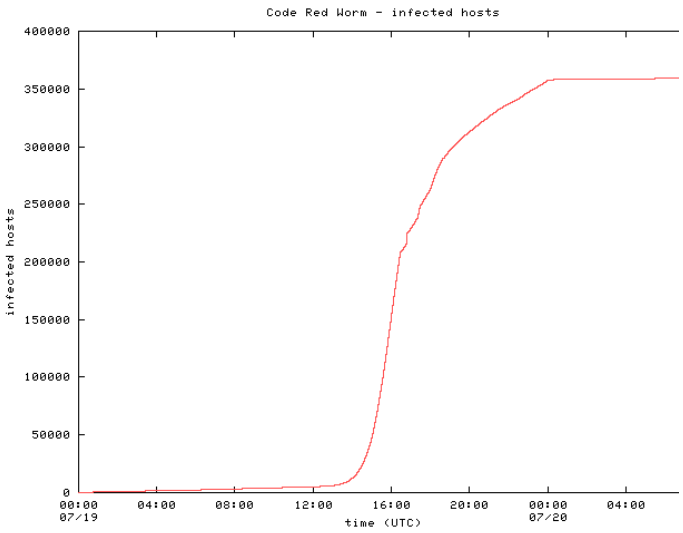


Abbildung 5.5: Anzahl der mit dem Wurm *Code Red v2* infizierten Hosts im Verlaufe der Zeit – Quelle: Originalgrafik aus [2]

Wurmbefall zu schützen. Abbildung 5.5 zeigt die Ausbreitung des Wurms „Code Red v2“ exemplarisch. Man kann hier den initial exponentiellen Anstieg der Anzahl der infizierten Rechner gut erkennen, bevor im Laufe der Zeit eine Sättigung einsetzt.

5.2.1.1 Code Red v2

Der Wurm Code Red v2 wurde am 19. Juli 2001 entdeckt [2]. Er ist als aktiver Wurm zu klassifizieren und infizierte über 359.000 Computersysteme in weniger als 14 Stunden, wie auch in Abbildung 5.5 zu sehen ist.

Bei Code Red v2 handelt es sich um eine Variante des Code Red Wurms, der eine Sicherheitslücke im *Microsoft Internet Information Server* (IIS) ausnutzt. Dabei kommt es zu einem Überlauf eines Pufferspeichers (*Buffer-Overflow*), der ausgenutzt werden kann, um fremden Code auf dem Zielsystem auszuführen.

Code Red v2 kann in zwei grundlegende Phasen eingeteilt werden: die Infektions- und die Angriffsphase. Während der Infektionsphase gilt es, möglichst viele Computersysteme zu infizieren. Dazu generiert eine Code-Red-v2-Instanz zufällig IP-Adressen und versucht, die entsprechenden Rechner zu infizieren. Es sind zwei Varianten des Wurms bekannt. Während eine Variante einen im Quellcode festgelegten *Seed*[1] zur zufälligen Generierung von IP-Adressen verwendet, nutzt die andere Variante einen zufälligen Seed. Bei der Variante mit zufällig generiertem Seed handelt es sich um eine Weiterentwicklung des Wurms. Ziel dieser Variation ist eine schnellere Ausbreitung durch eine gleichmäßigere Verteilung der generierten IP-Adressen im verfügbaren Adressraum. Dadurch wird eine bessere Abdeckung und Infektion zu erreichender und zu infizierender Systeme erzielt. An dieser Stelle sei nochmals auf Abbildung 5.5 verwiesen. In dieser Grafik ist die Anzahl der mit Code Red v2 infizierten Hosts gegen die Zeit aufgetragen. Es wird deutlich, in welch dramatischer Geschwindigkeit sich Code Red v2 ausbreiten konnte.

Ziel der Angriffsphase dieses Wurms ist eine verteilte *Denial-of-Service*-Attacke gegen `www1.whitehouse.gov`. Allerdings zielt die Implementierung des Wurms auf die IP-Adresse der Webseite und nicht auf den DNS-Eintrag. Es wird jedoch vor einem Angriff überprüft, ob TCP-Port 80 (Standard für Webserver) kurz vor der bevorstehenden Attacke aktiv ist. Durch eine derartige Schwäche in der Architektur von Code Red v2 ist es problemlos möglich, der

[1] Der *„seed“* (engl. für Saat, Samenkorn, Keim) bezeichnet den Startpunkt für einen Pseudozufallszahlen-Generator. Bei einheitlichem Seed erzeugt ein Pseudozufallszahlen-Generator immer die gleiche Folge an Zufallszahlen.

verteilten Denial-of-Service Attacke aus dem Weg zu gehen. Es gilt einzig für die Zeit des Angriffs den anzugreifenden Domainnamen mit einer anderen IP-Adresse zu assoziieren und unter der ursprünglichen IP-Adresse während des Angriffs keinen entscheidenden Server zu betreiben.

5.2.1.2 Code Red II

Code Red II wurde erstmals am 4. August 2001 entdeckt [12] und ist auch unter den Namen „CodeRed.v3“, „CodeRed.C“, „CodeRed III“, „W32.Bady.C“ und „CodeRed.F“ bekannt. Der Wurm nutzt zu seiner Verbreitung ebenfalls eine Sicherheitslücke im Microsoft Internet Information Server. Es handelt sich bei Code Red II um eine Variante des ursprünglichen Code Red Wurms, der am 16. Juli 2001 zu Tage trat [13].

Im Gegensatz zum originalen Code Red Wurm, der einen Denial-of-Service-Angriff auf die Webserver des Weißen Hauses verübte, ermöglicht Code Red II den vollständigen Zugriff auf ein befallenes System. Zu diesem Zwecke nutzt der Wurm eine bekannte Sicherheitslücke des Microsoft IIS in den Versionen 4.0 und 5.0. Es kommt dabei zu einem Pufferspeicherüberlauf (*Buffer-Overflow*), der durch ein *Exploit* (engl. für Ausbeute oder Heldentat) in der Datei `ldq.dll` verursacht wird. Im Falle einer Infektion eines Webservers vollzieht Code Red II eine Reihe an Operationen, die es ihm erlauben, das Trojanische Pferd „Trojan.VirtualRoot“ zu installieren und den infizierten Server neuzustarten.

Nach einer erfolgreichen Infektion beginnt der Wurm sich weiter zu verbreiten. Dazu wird die Spracheinstellung des befallenen Systems festgestellt. Im Falle einer chinesischen Sprache als Standardeinstellung werden 600, ansonsten 300 Threads erzeugt, die zufällig IP-Adressen generieren. Dabei werden mit einer Wahrscheinlichkeit von 1/2 das erste Byte und mit der Wahrscheinlichkeit 3/8 die ersten beiden Bytes der zu generierenden IP-Adresse durch die entsprechenden Bytes des infizierten Hosts ersetzt. Nur mit einer Wahrscheinlichkeit von 1/8 wird eine vollkommen zufällige IP-Adresse erzeugt. Grund für diese Vorgehensweise ist die Hoffnung, dass weitere Hosts im selben Netz des infizierten Rechners ebenfalls eine entsprechende Sicherheitslücke aufweisen und so ebenfalls infiziert werden können. Sollte ein weiterer noch nicht infizierter Server gefunden werden, der eine Weiterverbreitung des Wurms erlaubt, so wird dieses System ebenfalls von Code Red II befallen. Bei Code Red II handelt es sich entsprechend um einen aktiven Wurm, der sich über TCP-Port 80 verbreitet.

5.2.1.3 FreeBSD.Scalper

FreeBSD.Scalper, oder auch einfach „Scalper“, wurde erstmalig am 28. Juni 2002 entdeckt. Scalper ist auch unter den Namen „ELF_SCALPER.A“ und „BSD/Scalper.worm“ bekannt geworden [11]. Zu seiner Ausbreitung nutzt der Wurm eine Schwachstelle im Apache Web-Server, die einen Pufferspeicherüberlauf (*Buffer-Overflow*) zulässt. Diese Schwachstelle konnte bislang jedoch nur beim Betriebssystem FreeBSD ausgemacht werden.

Auch FreeBSD.Scalper fällt in die Klasse der aktiven Würmer. Zur Verbreitung überprüft Scalper einen Bereich an IP-Adressen, deren erstes Byte fest im Quellcode der Wurminstanz verankert ist. Das zweite Byte der IP-Adresse wird zufällig generiert, die verbleibenden zwei Bytes werden systematisch abgesucht. Für die Suche wird an jede generierte Adresse ein *HTTP GET request* verschickt. Sollte ein Apache Webserver gefunden werden, so versendet sich Scalper unter Ausnutzung der Schwachstelle des Servers selbst an das Zielsystem als Datei `/tmp/.uua` und führt sich selbst aus.

Durch Ausführung des Wurms wird der UDP-Port 2001 geöffnet, um auf weitere Instruktionen von außen entgegen zu nehmen. Diese Instruktionen können der folgenden Art sein:

- Sammle alle dem befallenen System bekannten E-Mail-Adressen.
- Betrachte sämtliche verfügbaren Webseiten.
- Versende Spam-Mails.
- Führe TCP, UDP oder DNS Floodings durch.

- Führe Befehle auf einer Shell aus.
- Weitere Denial-of-Service-Funktionen.

5.2.2 Erkennungsleistungen der einzelnen Koalitionspartner

Tabelle 5.2 fasst die Zeitpunkte der erstmaligen Erkennung der simulierten Wurmausbreitung durch die jeweils beteiligten GIDS-Partner zusammen. Der Beginn der simulierten Wurmausbreitung ist o. B. d. A. stets 0:00 Uhr.

Tabelle 5.2: Erkennungsleistung der einzelnen Kooperationspartner

Beginn der Wurmausbreitung:	Code Red II *00:00 Uhr*	Scalper *00:00 Uhr*
DFN-CERT	—	00:28:02 Uhr
LMU	—	01:07:13 Uhr
LRZ	00:09:16 Uhr	00:21:40 Uhr
RRZN	00:12:26 Uhr	00:21:59 Uhr

Durch die Ausbreitungsstrategie der Wurminstanzen bedingt (vgl. hierzu Abschnitt 5.2.1) ist eine Erkennung der Wurmausbreitung genau dann recht einfach, wenn das erste System im eigenen Netz befallen wurde oder auch lediglich versucht wurde, das System zu befallen (abhängig von der Strategie des Wurms zu seiner Verbreitung). Dennoch ist eine Site-lokale Erkennung natürlich auch von der Menge der verfügbaren Daten abhängig, die wiederum implizit von der Größe des beobachteten Netzes abhängt. Im Fall von GIDS stehen beim DFN-CERT sowie an der LMU lediglich IPv4 /24-Netze, beim LRZ und RRZN hingegen /16-Netze unter Beobachtung.

Im Falle von Code Red II führt die unterschiedliche Größe der eigenen Netze dazu, dass das DFN-CERT und die LMU München die Wurmausbreitung alleine nicht erkennen konnten. Das LRZ und RRZN hingegen waren in der Lage mit Hilfe der Site-lokalen GIDS-Komponenten die Wurmausbreitung zu entdecken.

Bedingt durch die Ausbreitungsstrategie von Scalper ist die Erkennung durch ein IDS leicht, wenn man beginnt, IDS-relevante Ereignisse zu korrelieren. Ein mit Scalper infiziertes System generiert zufällig die ersten drei Bytes einer IPv4-Adresse und versucht systematisch durch sequentielles Durchwandern des Wertebereichs des letzten Bytes alle verfügbaren Systeme zu infizieren (vgl. auch Abschnitt 5.2.1.3). Entsprechend gut ist auch die in Tabelle 5.2 dargestellte Erkennungsleistung aller Kooperationspartner bereits im Einzelfall. Im Hinblick auf die Effizienz der Erkennung ist festzustellen, dass jeder der Partner in der Lage war, bereits beim ersten auftretenden Infektionsversuch die Wurmausbreitung festzustellen.

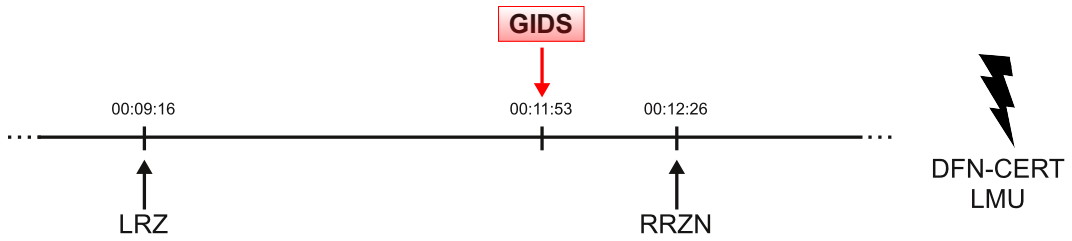
5.2.3 Kooperativen Erkennungsleistung

In Tabelle 5.3 ist die Erkennungsleistung des GIDS in seiner Gesamtheit dargestellt. Auch hier gilt, der Beginn der simulierten Wurmausbreitung ist o. B. d. A. der Einfachheit halber 0:00 Uhr. Durch das Zusammenführen aller anfallenden Alarmmeldungen zu den jeweiligen Wurmausbreitungen ergibt sich eine Flut an Daten, die durch die zentrale IDS-Komponente des GIDS zu verarbeiten sind, was aber selbst innerhalb der Testumgebung für Entwicklungszwecke klaglos funktioniert hat.

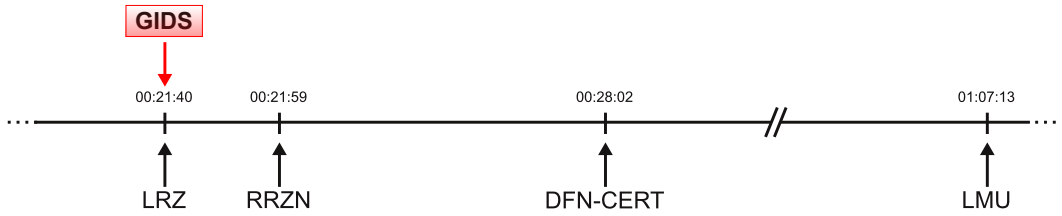
Im Falle von Code Red II ist erfreulich, dass der Wurm erfolgreich erkannt werden konnte. Im Sinne der Effektivität ist festzustellen, dass sich für vergleichsweise kleine (im Sinne der von ihnen betriebenen Netzgröße) GIDS-Partner erst durch den Zusammenschluss und die kooperative Erkennung der potentielle Wurmbefall überhaupt hat erkennen lassen. Im Sinne der Effizienz hingegen ergibt sich ein geteiltes Bild. Der Vergleich in Tabelle 5.2 zeigt, dass das LRZ alleine Code Red II früher erkennen konnte und somit effizienter als das GIDS-Gesamtsystem gewesen ist. Im Falle des RRZN hat sich ein kleiner Effizienzgewinn ergeben, der kaum erwähnenswert ist.

Tabelle 5.3: Kooperative Erkennungsleitung

	Code Red II	Scalper
Beginn der Wurmausbreitung:	*00:00 Uhr*	*00:00 Uhr*
GIDS (gesamt)	00:11:53 Uhr	00:21:40 Uhr



(a) Code Red II



(b) Scalper

Abbildung 5.6: Erkennungsleitung von GIDS im Einzel- und Kooperationsbetrieb

Im Falle von Scalper war die Erkennungsleitung der einzelnen Kooperationspartner im Hinblick auf Effektivität und Effizienz bereits exzellent. Aus Tabelle 5.3 geht hervor, dass auch die kooperative Erkennungsleitung von GIDS für den Fall von Scalper optimal ist. Es ist die frühest mögliche Erkennung der Wurmausbreitung erfolgt und somit eine optimale Frühwarnung möglich.

5.2.4 Zusammenfassung

Abbildung 5.6 stellt die Erkennungsleistungen von GIDS zusammenfassend graphisch dar. Aus Abbildung 5.6(a) geht dabei die Erkennungsleistung, sowohl der einzelnen Kooperationspartner, als auch die Erkennungsleistung des GIDS im kooperativen Betrieb bei der Einbringung einer simulierten Wurmausbreitung einer Code-Red-II-Instanz hervor.

Es fällt auf, dass sowohl das DFN-CERT als auch die LMU nicht in der Lage waren, die Wurmausbreitung von Code Red II seperat zu erkennen. Das LRZ und RRZN hingegen haben eine Erkennung auch im Alleingang erfolgreich durchführen können. Der Grund hierfür liegt in der Strategie zur Ausbreitung des Wurms (vgl. Abschnitt 5.2.1.1) in Zusammenhang mit den jeweiligen Netzgrößen der einzelnen Partner begründet. Während die LMU und das DFN-CERT nur vergleichsweise kleine Netzbereiche zur Beobachtung heranziehen können, verfügen das LRZ und RRZN über deutlich größere IP-Adressbereiche, die sie analysieren können. Code Red II versucht mit gewissen Gewichten belegte, aber dennoch zufällig generierte IP-Adressen zu infizieren, was natürlich eine Erkennung in größeren Netzabschnitten erleichtert.

Es ist anzumerken, dass im Falle der simulierten Code Red II Ausbreitung alle Simulationsdaten in nicht anonymisierter Form eingeflossen sind. Dies hat durch die oben beschriebene spezielle Ausbreitungsstrategie zur Folge, dass dies einer Erschwerung der Angrifferkennung

gleich kommt. Bei einer Anonymisierung durch die GIDS-Komponenten würde stets das letzte Byte einer angegriffenen IPv4-Adresse auf Null gesetzt werden, was naheliegenderweise eine Korrelation der Ereignisse vereinfacht. Das Aussetzen der Anonymisierung im Falle genau dieses Simulationslaufs ist bewusst gewählt worden, um die Erkennung zu erschweren und die Leistungsfähigkeit von GIDS besser bewerten zu können.

Führt man alle simulierten Meldungen zur Code Red II Ausbreitung im Rahmen von GIDS zusammen, so ist man nach wie vor in der Lage die Wurmausbreitung zu erkennen. Der Erkennungszeitpunkt liegt hierbei zwischen der Erkennung durch das LRZ und das RRZN, was im kooperativen Fall durch das erhöhte Grundrauschen an Alarmmeldungen und den erheblich erweiterten Blick auf eine größere Anzahl von Netzabschnitten liegt. Dennoch ist insbesondere für die kleineren Partner von GIDS ein erheblicher Zugewinn zu verzeichnen.

Abbildung 5.6(b) stellt analog die Erkennungsleistung bei simulierter Ausbreitung von FreeBSD.Scalper dar. Da sich Scalper recht leicht erkennbar ausbreitet (vgl. Abschnitt 5.2.1.3), waren alle GIDS-Partner bereits bei der Erkennung der Wurmausbreitung in Eigenregie erfolgreich. Auch hier hat das GIDS-Gesamtsystem die optimale Erkennungsleistung vollbracht und Scalper im kooperativen Betrieb zum frühest möglichen Zeitpunkt erkannt.

Kapitel 6

Zusammenfassung

In dem vorliegenden Meilenstein 36 wurde das GIDS-Produktivsystem beschrieben, das die Grundlage für einen GIDS-Dienst bietet. Technisch basiert dieses System auf der prototypichen Implementierung des GIDS, die im Detail in [8] beschrieben worden ist. Um die Leistungsfähigkeit des GIDS zu demonstrieren, wurde ein Nachweis der Tragfähigkeit erbracht, der die bereits erkannten Angriffe im Einsatz zusammenfasst. Im Rahmen der Bedrohungsanalyse in der Startphase des Projektes wurden Internet Würmern als wichtiges Szenario erkannt. Da in der Testphase keine Ausbrüche neuer Würmer aufgetreten sind, wurde eine Simulation durchgeführt, bei der simulierte Internet-Würmer erfolgreich erkannt werden konnten.

Die technischen Erweiterungen haben sich an den Anforderungen eine produktiven Dienstes orientiert. Dabei sind die Ergebnisse der Kalibrierung der Sensorik, die Paketierung der Komponenten auf der Seite des Betreibers und der Ressourcenzulieferer eingeflossen. Des Weiteren wurde der Funktionsumfang des GIDS-Portals erweitert und dies an die Struktur des D-Grids angepasst.

Organisatorisch wurde ein Konzept zum Betrieb des GIDS vorgestellt, das die Grundlage für einen Betrieb bildet und für einen kommerziellen Dienst entsprechend erweitert werden kann. Für einen reibungslosen Betrieb sind Anleitungen für die Anbindung einer neuen administrativen Domäne und für die Benutzung der GIDS-Portals vorhanden.

Ein weiterer wichtiger Punkt für das GIDS und dessen Verwertung hat sich mit der Kooperation mit dem vom BSI und CERT-Verbund geförderten Frühwarnsystem CarmentiS ergeben. Mit der Integration eines Datenexportes nach CarmentiS fliessen diese Daten in das Lagebild ein, das durch das Frühwarnsystem erstellt wird. Die Kooperation ist für alle GIDS-Partner optional und ermöglicht nach Unterzeichnung einer Kooperationsvereinbarung den Zugriff auf das aktuelle Lagebild. Davon profitieren sowohl CarmentiS als auch das D-Grid. Des Weiteren lassen sich die Daten für die Bearbeitung von Sicherheitsvorfällen nutzen. So bietet das DFN in Zusammenhang mit dem DFN-CERT einen Dienst an, der Daten über kompromittierte System sammelt und diese an betroffene Seiten im DFN weitergibt. Die Kooperation mit CarmentiS bietet als zusätzlichen Vorteil die Möglichkeit, die GIDS-Daten für diesen Zweck nutzen zu können.

Anhang A

Anleitung zur Bedienung des GIDS-Portal

Ein Grid-basiertes, föderiertes Intrusion Detection System zur Sicherung der D-Grid Infrastruktur (GIDS)

Anleitung zur Bedienung des GIDS-Portals

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

In diesem Dokument wird die Verwendung des Portals des Grid-IDS beschrieben. Der Zweck des Portals ist die Darstellung und Übersicht über die IDS-Meldungen, die von Sensoren des Grid-IDS (GIDS) erzeugt und an den Betreiber weitergeleitet wurden.

A.1 Einleitung und Übersicht

Mit Grid-IDS (GIDS) wird dem D-Grid ein System zur Verfügung gestellt, um Einbrüche auf Systeme im Grid zu erkennen. Die Darstellung der Alarmmeldungen erfolgt durch ein webbasiertes Portal. In diesem Dokument wird die Benutzung des Portals beschrieben, welches sich sowohl an Mitglieder einer VO als auch an Administratoren von Grid-Ressourcen richtet. Das Portal kann von jedem Teilnehmer des D-Grids mit gültigem GridCA-Zertifikat benutzt werden. Die Berechtigungen hängen im Einzelnen von den Rollen im D-Grid (VO-Mitglied oder Administrator) ab.

Dabei wird zwischen den folgenden Benutzergruppen unterschieden:

VO-Mitglieder sind mit zwei Einschränkungen in der Lage, auf IDS-Meldungen zuzugreifen. Zunächst sind nur Meldungen sichtbar, die von Ressourcen stammen, auf die Zugriff für die VO besteht. Weiterhin lassen sich die Zugriffsrechte von den für die Ressourcen zuständigen Administratoren einschränken.

Administratoren von Grid-Ressourcen haben Zugriff auf alle IDS-Meldungen, die aus der oder den eigenen Ressourcen stammen. Weiterhin können diese Zugriffsrechte für VOs vergeben.

GIDS Betreiber können auf alle IDS-Meldungen zugreifen. Dies ermöglicht dem Betreiber die globale Sicht auf alle Meldungen („Lagebild“). Dadurch können ungewöhnliche oder sogar bedrohliche Zustände erkannt werden, die auf eine hohe Gefährdung des D-Grids oder einzelner Ressourcen hinweist. Allerdings werden nur Meldungen und zugehörigen Daten im Portal angezeigt, die nicht durch die Ressourcenprovider gefiltert wurden.

A.1.1 Erkennung von Angriffen und Frühwarnung

Angriffe können im Portal anhand der korrelierten IDS-Meldungen und der verschiedenen Statistiken erkannt werden. Dies wird dadurch unterstützt, dass das Portal IDS-Meldungen sämtlicher am GIDS beteiligten Partner erhält und darstellt. Es ist also eine globale Sicht auf die aktuelle Sicherheitssituation des D-Grids möglich. Zwar werden aus Gründen des Datenschutzes die Daten zumindest teilweise anonymisiert, jedoch bleiben die folgenden Vorteile:

- Es lassen sich Gemeinsamkeiten und Unterschiede bei den IDS-Meldungen der beteiligten Seiten untersuchen. Findet ein Angriff auf mehrere Partner des D-Grids statt, spiegelt sich dies in Gemeinsamkeiten bei den erkannten Angriffen wider. Dies führt zu einer Korrelation bezüglich des Auftretens und der Häufigkeit bestimmter IDS-Meldungen.

- Im GIDS-Portal kann ein Lagebild des D-Grids dargestellt werden. Aufgrund der hohen Aggregation und Anzahl der IDS-Meldungen kann davon ausgegangen werden, dass die statistischen Schwankungen geringer als bei den einzelnen Partnern im D-Grid sind. Größere Schwankungen (Anomalien) können deshalb auf eine erhöhte Gefährdungslage (höhere Anzahl von Angriffsmeldungen hinweisen.

- Ein Nachweis für eine erhöhte Bedrohungslage kann die Grundlage für die Verarbeitung personenbezogener Daten liefern. So bietet das Datenschutzgesetz Normen für die Beseitigung von Störungen.

Die Erkennung von Angriffen kann sowohl durch einen GIDS-Partner als auch dem Betreiber erfolgen. Betreiber von Ressourcen oder Mitglieder von VOs haben Zugriff auf die Daten der eigenen Ressourcen und können dort kritische Zustände aufgrund von IDS-Meldungen

oder Statistiken erkennen. Eine erweiterte Analyse kann durch den Beteiber oder Ressourcen-Administratoren erfolgen. Letztere haben die Möglichkeit, den Zustand in dem privaten Datenbestand zu analysieren. Ist der Angriff charakterisiert, bieten diese Informationen die Grundlage für Frühwarnung im GIDS. Der GIDS-Betreiber kann im globalen Datenbestand nachvollziehen, ob andere Seiten in den Vorfall verwickelt sind und diese entsprechend warnen.

A.2 Bedienung des Portals

In diesem Abschnitt wird die Bedienung des GIDS-Portals für Mitglieder des D-Grids beschrieben. Dies betrifft die Anmeldung und Beschreibung der Sichten des Portals.

A.2.1 Anmelden am Portal

Die Anmeldung am Portal erfolgt ausschließlich über HTTPS, um sowohl den Server als auch Client zu authentifizieren und eine sichere Verbindung zu gewährleisten. Für die Anmeldung ist ein gültiges D-Grid-Zertifikat notwending. Weiterhin müssen dessen Daten in der GRRS-Datenbank hinterlegt sein, um autorisierten Zugriff auf die Daten zu erhalten.

A.2.2 Sichten des Portals

Unabhängig von der jeweiligen Sicht auf das Portal befindet sich auf der linken Seite das Hauptmenü, durch das die jeweiligen Sichten navigiert werden können. Die Details in den Sichten sind von von den jeweiligen Privilegien des Benutzers abhängig. Dies beinhaltet die Rollen eines Administrators einer Grid-Ressource und die Mitgliedschaft in einer VO.

Startbildschirm

Nach der Anmeldung erscheint die personalisierte Startseite des Portals. Dabei werden zwei Tabellen angezeigt, die sich auf Grid-Ressourcen und VOs beziehen. Wie bereits vorher beschrieben werden die Informationen aus der GRRS-Datenbank entnommen. In der ersten Tabelle werden die Rollen des angemeldeten Benutzers für die Grid-Ressourcen dargestellt. Dabei wird zwischen den folgenden Rollen unterschieden:

User haben Zugriff auf die IDS-Meldungen, deren Ursprung ein IDS in der entsprechenden Ressource ist.

Admins können zusätzlich zum Zugriff auf die Daten ihrer Ressource die Berechtigungen für VOs setzen. Wird der Menüpunkt „configure“ ausgewählt, werden alle VOs aufgelistet, die für die Benutzung der Ressource berechtigt sind. Per Auswahl lässt sich auswählen, ob die ausgewählte VO berechtigt ist, die IDS-Meldungen aus der eigenen Ressource sehen zu können.

Weiterhin werden in der darunter befindlichen Tabelle die VOs aufgelistet, in denen der angemeldete Benutzer aufgelistet ist.

Anzeige der IDS-Meldungen

Die Sicht „Current Alerts“ listet alle korrelierten IDS-Meldungen auf, auf die der angemeldete Benutzer berechtigten Zugriff hat. Hierbei ist zu beachten, dass nur **korrelierte** IDS-Meldungen angezeigt werden. Diese werden durch den Prelude-Correlator erzeugt, wenn entweder mehrere IDS-Meldungen in kurzer Zeitfolge auftreten, oder kritische IDS-Meldungen erzeugt werden.

In der ersten Zeile über den Meldungen können Einschränkungen bezüglich der Ressourcen, des Zeitraums und der Priorität (Impact) vorgenommen werden. Weiterhin besteht die

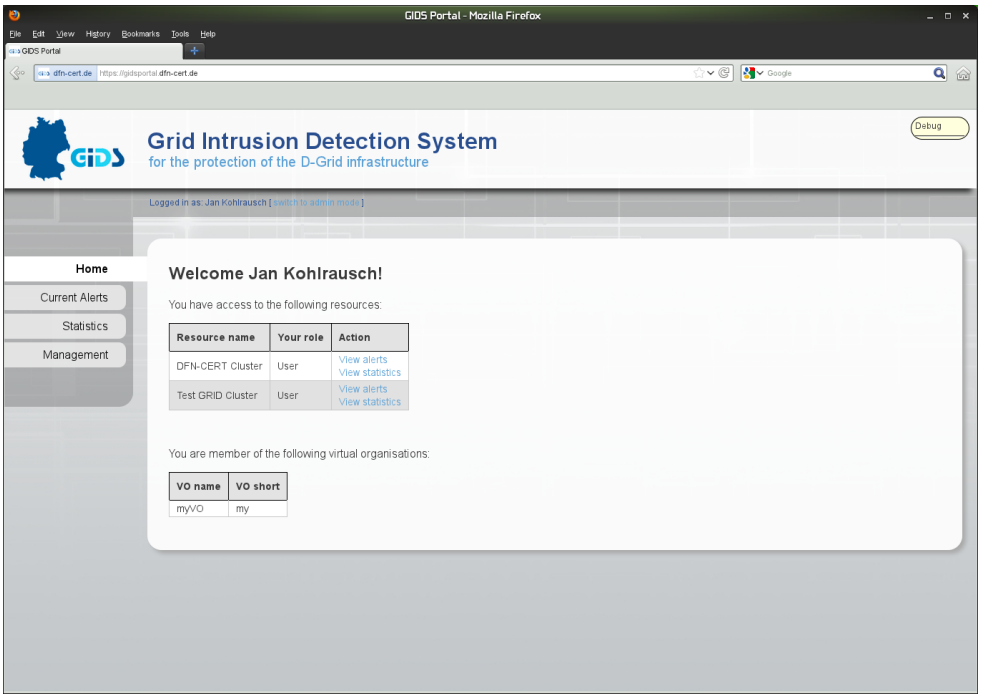


Abbildung A.1: Ansicht der Statistiken im GIDS-Portal

Möglichkeit nach Präfixen von IP-Adressen oder Bezeichnungen des Alarmes zu suchen. Standardmäßig werden die Meldungen chronologisch nach dem Zeitpunkt des Aufzeichnens aufgelistet. Die Zeile direkt über den Meldungen beinhaltet Felder, um die Reihenfolge der Sortierung zu ändern.

Eine IDS-Meldung wird über mehrere Spalten dargestellt, die die folgende Bedeutung haben:

Alert ID. Die ID der IDS-Meldung. Durch Anwählen der ID werden weitere Informationen der entsprechenden Meldung angezeigt.

Analyzer Time. Der Zeitpunkt, an dem die Meldung erzeugt wurde.

Alert Name und Classification. Der Name und die Klassfizierung geben an, um welche Art von Alarm es sich handelt. Der Name leitet sich von der Komponente ab, die den Alarm erzeugt hat; dies ist im GIDS das Plug-in des Correlators. Die Klassfizierung bezeichnet die Klasse, die dem Alarm zugeordnet wurde; dies kann beispielsweise die Klasse „Portscan“ oder „Brute-Force-Angriff“ sein.

Severity. Die Severity ist eine Einschätzung für das Schadensausmaß des Alarms. Grundlage ist zum einen die Bewertung durch das IDS, das den Alarm aufgezeichnet hat, und zum anderen die Bedrohungsanalyse des GIDS-Projektes.

Ressource. Die ID der Ressource.

Anzeige statistischer Daten

Während die Sicht der IDS-Meldungen die Alarm-Details anzeigt, dient die Sicht „Statistics“ zur Darstellung der globalen Lage (Lagebild) des D-Grids oder einzelner Ressourcen. Dabei werden alle IDS-Meldungen und nicht nur die korrelierten Meldungen berücksichtigt. Aufgrund der hohen Aggregation kann davon ausgegangen werden, dass die statistischen Schwankungen relativ gering sind. Größere Schwankungen (Anomalien) können deshalb entweder auf eine

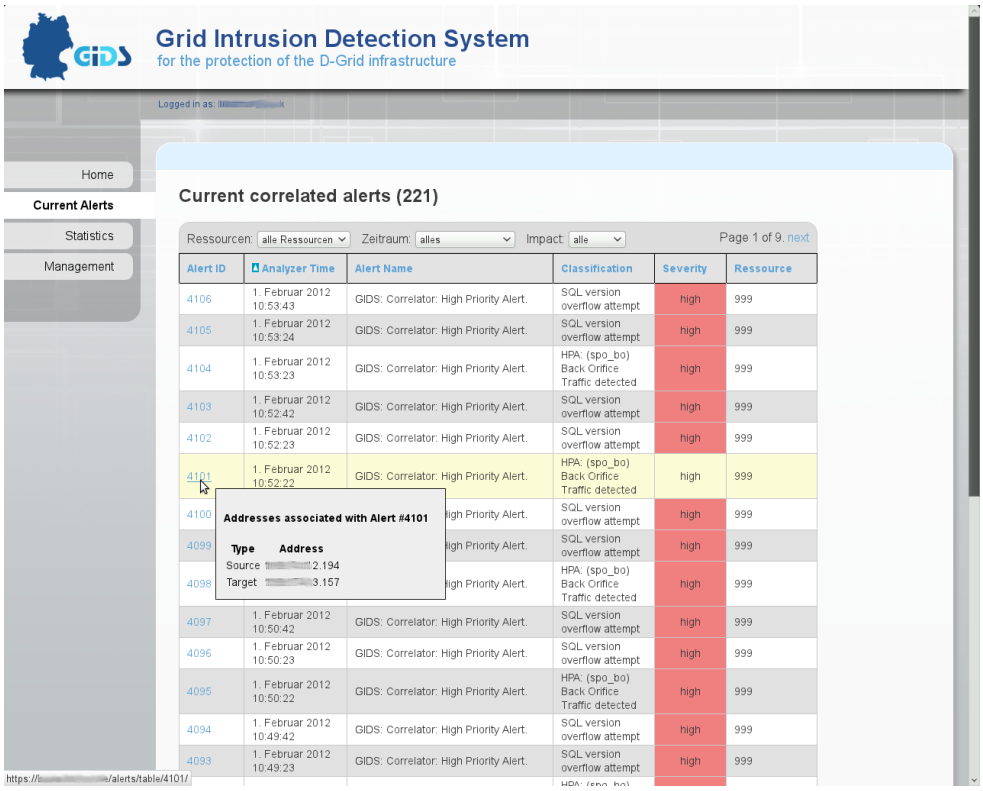


Abbildung A.2: Ansicht der Statistiken im GIDS-Portal

erhöhte Gefährdungslage (höhere Anzahl von Angriffsmeldungen) oder auf ein fehlkonfiguriertes Netz oder System hinweisen.

Die Auswahl der Ressourcen erfolgt direkt unterhalb der Überschrift „Global Statistics“ mittels „ShowHide“ (siehe Abb. A.3). Hier können einzelne Ressourcen oder das Lagebild „alle Ressourcen“ ausgewählt werden. Letzteres bezieht sich auf alle Ressourcen des D-Grids unabhängig davon, ob der Benutzer Zugriff auf diese hat oder nicht. Da nur hochgradig aggregierte Daten ohne Personenbezug angezeigt werden, verletzt dies nicht das Datenschutzkonzept. Der Zweck dieser Sicht ist es, ein globales Lagebild des D-Grids zu erstellen, dass eine Einschätzung der aktuellen Gefährdungslage wiedergibt.

In der ersten Grafik (A.4) werden die Statistik der am häufigsten aufgetretenen IDS-Meldungen angezeigt. Hier lassen sich gut fehlkonfigurierte Systeme oder IDS erkennen.

Die zweite Statistik (A.5) gibt eine Übersicht der Verteilung des Schweregrades („Impact“) der IDS-Meldungen. Grundlage hierfür bietet die Klassifizierung, die von jeweiligen IDS vergeben wurde. Verschiebt sich der Schwerpunkt hin zu den IDS-Meldungen mit hohem Impact, kann dies auf einen Angriff auf das D-Grid oder die einzelne Ressource hinweisen. In diesem Fall ist eine Analyse der einzelnen IDS-Meldungen durchzuführen.

In der folgenden Statistik werden nur die IDS-Meldungen mit hohem Impact berücksichtigt und nach Häufigkeit sortiert aufgelistet. Motivation ist die Erkennung einer Häufigkeit oder Anomalie bei den kritischen IDS-Meldungen.

Die vierte Statistik gibt eine Übersicht über die Ports, die Ziel der Angriffe sind. Auch hier ist das Ziel, Anomalien bei den Meldungen erkennen zu können.

Management

Die Management Sicht erfüllt die folgenden Aufgaben: Zuerst wird gezeigt, auf welche Ressourcen der angemeldetet Benutzer administrativen Zugriff hat. Grundlage dafür ist, dass der Benutzer in der GRRS-Datenbank des D-Grids als Administrator der entsprechenden Res-

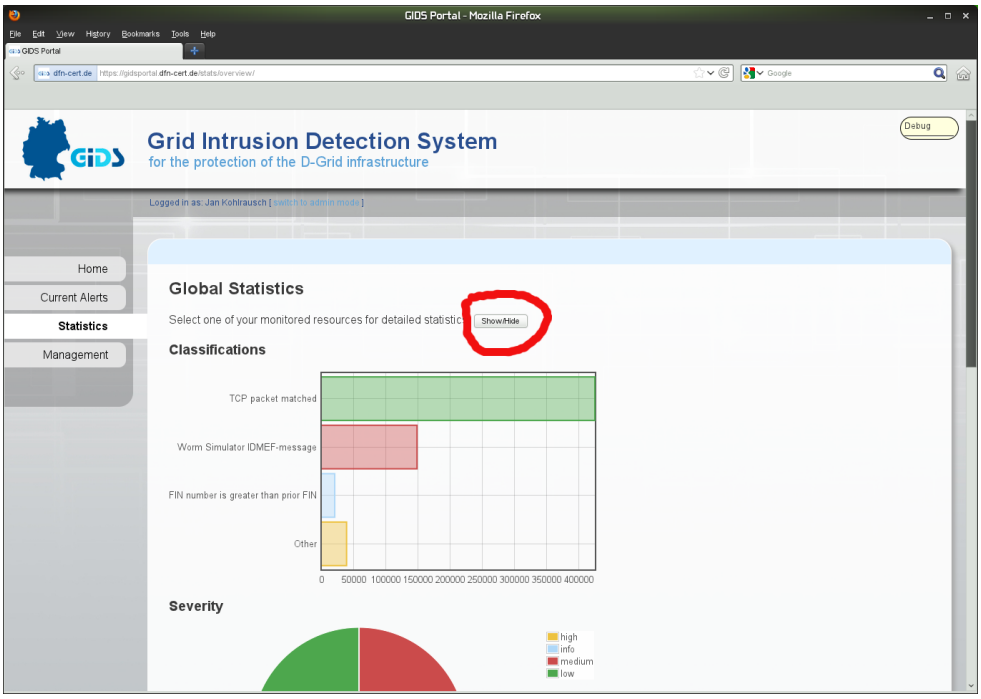


Abbildung A.3: Ansicht der Statistiken im GIDS-Portal

source oder Ressourcen gelistet ist (Abb. A.8). Über die „configure“ Aktion lässt sich die entsprechende Ressource auswählen. Wie in Abb. A.10 gezeigt, werden die VOs aufgelistet, die laut Angaben der GRRS-Datenbank zu deren Nutzung berechtigt sind. Dies berechtigt die VO implizit, IDS-Meldungen zu der Ressource zu sehen. Ist dies durch den Administrator nicht gewünscht, kann der Zugriff durch Deaktivierung verhindert werden. In Abb. A.10 kann zum Beispiel nur die VO „kerndgrid“ auf Meldungen aus der Ressource zugreifen.

Die folgenden zwei Tabellen (Abb. A.9) ermöglichen es, Warnmeldungen per E-Mails bezüglich der Zustandes von Ressourcen zu konfigurieren. Die Meldungen werden an die Adressen versendet, die in der ersten Tabelle eingetragen wurden. Dabei wird pro E-Mail Adresse angegeben, ob der Export aktiv ist und welcher Filter mit dem Export verknüpft werden soll (siehe Abb. A.11). Die Filter werden in der unteren Tabelle angelegt und konfiguriert.

A.3 Datenschutz

Da in dem GIDS-Portal detailierte Informationen über Angriffe angezeigt werden, fällt es auch unter das Bundesdatenschutzgesetz (BDSG). Wie auch im Datenschutzkonzept des GIDS beschrieben, werden die Daten per Default vor dem Export anonymisiert. Dies betrifft insbesondere die IP-Adressen der Quelle und des Ziels des Angriffs. Allerdings ist das Portal technisch dazu ausgelegt, die vollständigen Informationen der Meldungen anzuzeigen. Dazu muss allerdings eine gesetzliche Norm erfüllt sein (z. B. das Einverständnis des Besitzers der Daten oder eine Genehmigung zur Beseitigung einer Störung). Aufgrund der Autonomie der Ressourcen-Provider haben diese jedoch letztendlich die Entscheidung, welche Daten in welcher Form exportiert werden.

Weiterhin fällt die Entscheidung, für wen die Daten sichtbar sind unter den Einfluss des Datenschutzkonzeptes. Hierbei wird das Prinzip „least Privilege“ umgesetzt. Die Ressourcenprovider und Mitglieder einer VO haben nur auf die Daten der eigenen Ressourcen Zugriff. Um jedoch die Anforderungen an die Erkennungsleistung zu erfüllen, hat der Betreiber als zentrale Instanz Zugriff auf alle Daten. Dies ist notwendig, um Korrelationen zwischen den Daten der Ressourcenprovider zu finden und globale Anomalien im D-Grid zu erkennen. Da

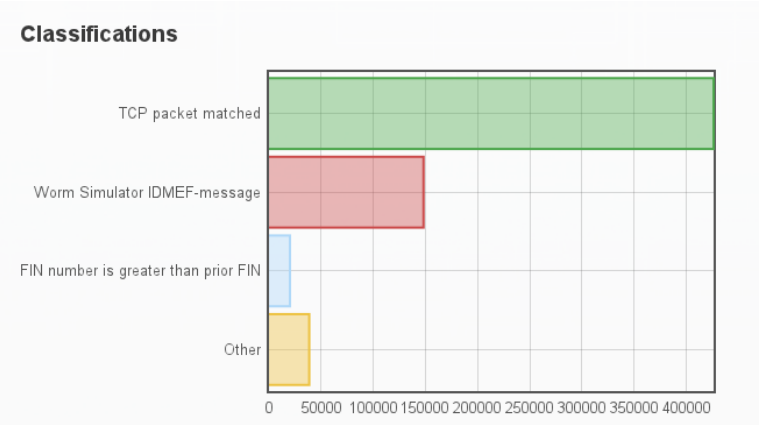


Abbildung A.4: Ansicht der Statistiken im GIDS-Portal

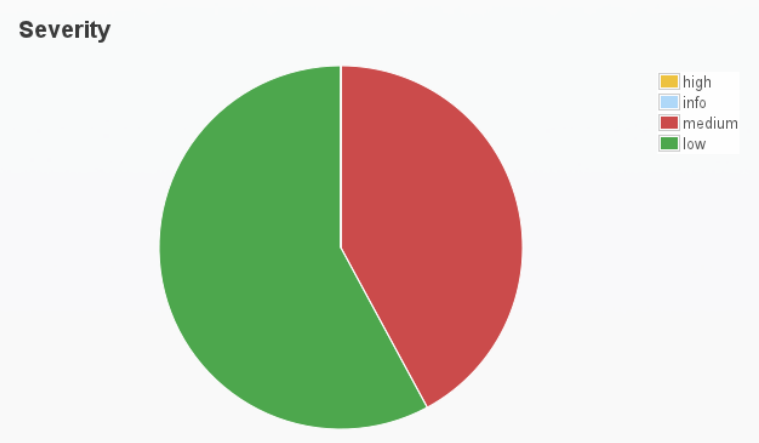


Abbildung A.5: Ansicht der Statistiken im GIDS-Portal

das Portal jedoch per Default nur aggregierte oder anonymisierte Daten umfasst, die keinen Personenbezug zulassen, ist dies konform zum Datenschutz.

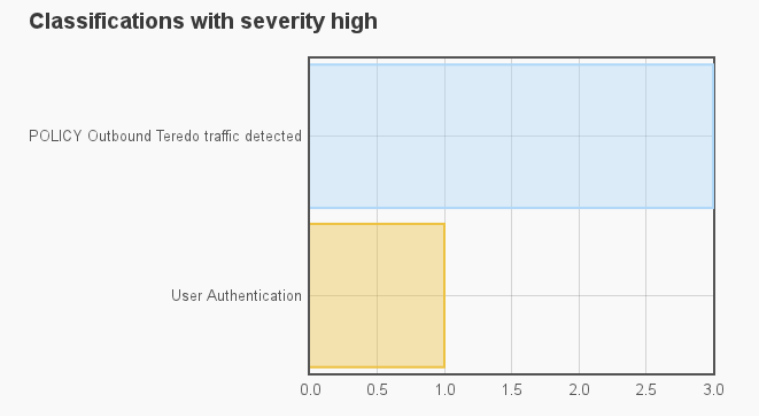


Abbildung A.6: Ansicht der Statistiken im GIDS-Portal

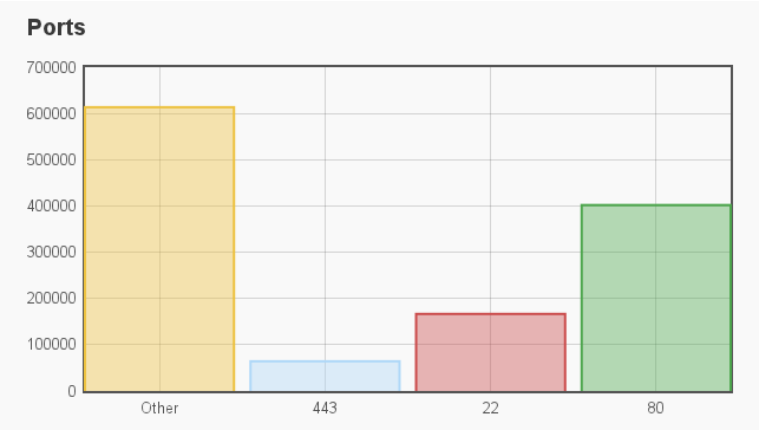
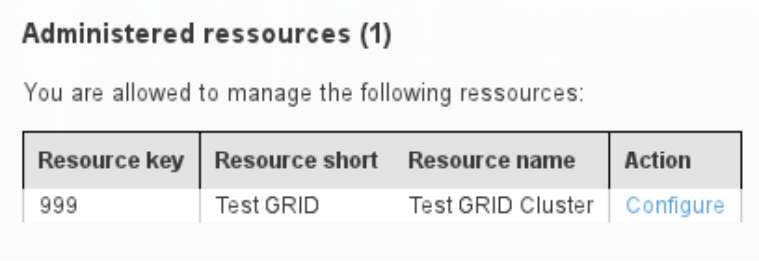


Abbildung A.7: Ansicht der Statistiken im GIDS-Portal



Administered ressources (1)

You are allowed to manage the following ressources:

Resource key	Resource short	Resource name	Action
999	Test GRID	Test GRID Cluster	Configure

Abbildung A.8: Ansicht der Statistiken im GIDS-Portal

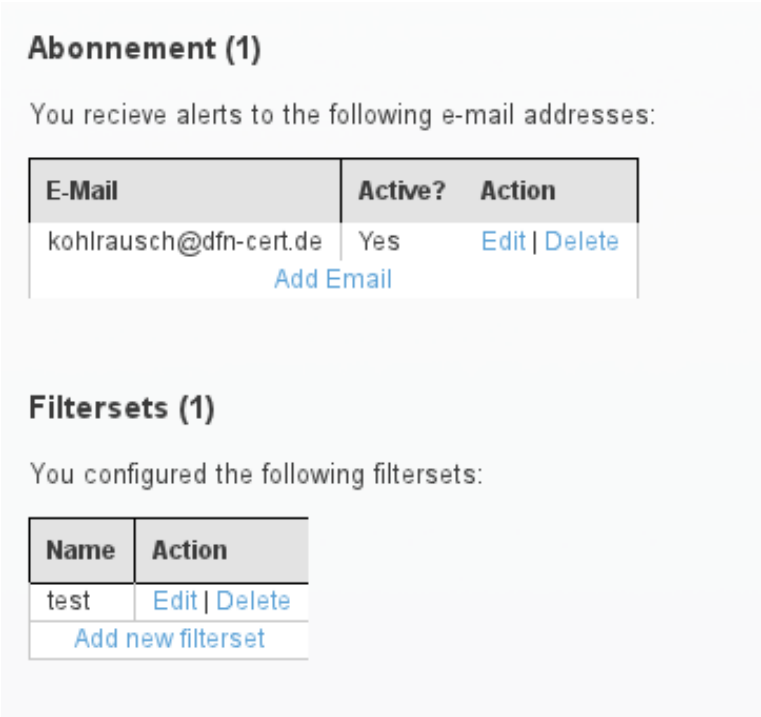


Abbildung A.9: Ansicht der Statistiken im GIDS-Portal

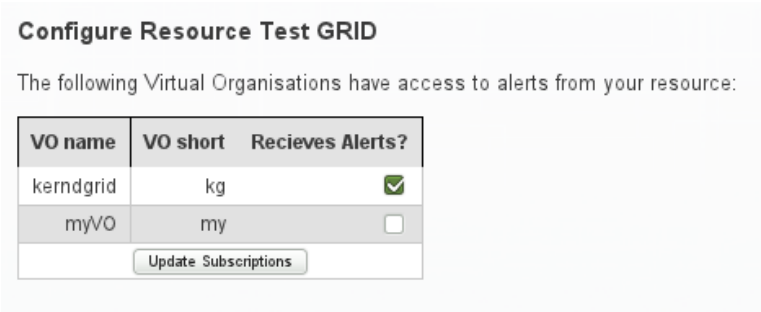


Abbildung A.10: Ansicht der Statistiken im GIDS-Portal

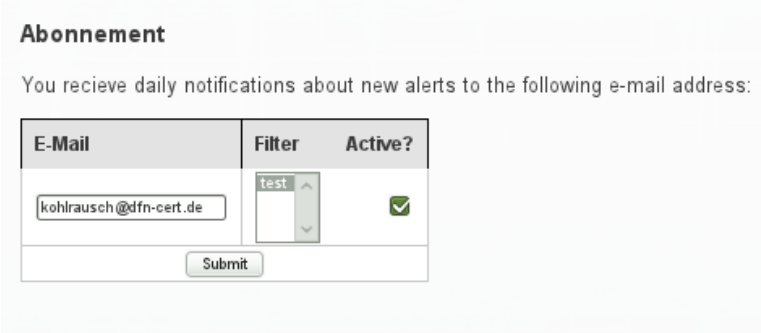


Abbildung A.11: Ansicht der Statistiken im GIDS-Portal

Anhang B

Anleitung zum Anbinden einer administrativen Domäne an das GIDS

Ein Grid-basiertes, föderiertes Intrusion Detection System zur Sicherung der D-Grid Infrastruktur (GIDS)

Anleitung zur Installation der GIDS-Komponenten bei D-Grid-Ressourcenanbietern

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Dieses Dokument beschreibt schrittweise das Vorgehen, um eine administrative Domäne an das GIDS anzuschließen. Dies betrifft die Basisinstalltion eines `gids-client` inkl. einer Standardkonfiguration und die IDS-Sensorik. Es hat sich gezeigt, dass Debian Linux im Gegensatz zu anderen Linux Distributionen die Paketeabhängigkeiten zur Installation der GIDS-Pakete sehr gut unterstützt. Aus diesem Grund wird die Verwendung von Debian Linux für die Installation des GIDS-Clients empfohlen. Es ist wichtig zu beachten, dass dies ausschließlich die Installation des GIDS-Clients betrifft. Die anderen Systeme auf denen die IDS-Sensoren laufen, sind davon nicht betroffen; hier kann die alte Infrastruktur übernommen werden.

Beispielhaft werden die Befehlsfolgen anhand von *Debian Linux „stable“* auf einem 64-Bit-System gezeigt. Beachten Sie, dass Sie für die meisten der nachfolgend genannten Befehle `root`-Rechte benötigen, um sie erfolgreich auszuführen.

B.1 Einleitung

In diesem Dokument wird die Einbindung einer administrativen Domäne in das Grid-IDS (GIDS) beschrieben, wie sie in Abb. B.1 gezeigt ist. Ein wichtiges Ziel von GIDS ist die Autonomie der Partner. Aus diesem Grund kann die Auswahl der Sensorik und die Anonymisierung und Filterung der Daten frei gewöhlt werden. Allerdings ist die technische Anforderung vorhanden, dass die Sensorik das Prelude-Rahmenwerk unterstützt oder daran angepasst ist. Die Einbundung beinhaltet in der Übersicht die folgenden Schritte:

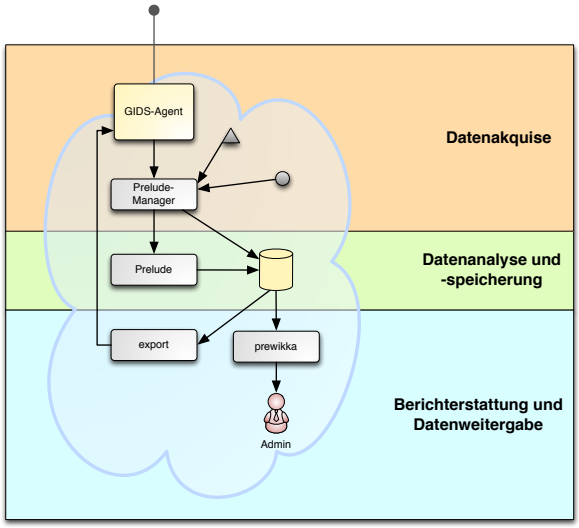


Abbildung B.1: Überblick über die Komponenten auf Seiten des Ressourcenproviders

Registrierung beim GIDS-Projekt. Der erste Schritt ist die Registrierung beim GIDS-Betreiber. Dieser stellt ein digitales Zertifiket zur Verfügung, über das Daten an das GIDS gesendet und empfangen werden können.

Aufsetzen des GIDS-Clients. Danach erfolgt das Aufsetzen des GIDS-Client. Dieser stellt die Verbindung zwischen der Sensorik und dem GIDS-Bus her. Der GIDS-Clients kann entweder auf einem virtuellen oder physikalischen System installiert werden. Da sich Debian Linux für den Betrieb des Client bewährt hat, wird diese Distribution empfohlen. Weiterhin werden alle benötigten Softwarepakete für Debian zur Verfügung gestellt.

Aufsetzen oder Einbundung der Sensorik. Es können vorhandene Sensoren eingebunden werden oder neue aufgesetzt werden. Die Sensorik wird über das Prelude-Rahmenwerk an den GIDS-Client gebunden. Dabei übernimmt der GIDS-Client die Rolle des Prelude-Managers und die Sensorik die Rolle der Prelude-Clients.

Aufsetzen des Prelude-Correlators. Der Prelude-Correlator ist ein Regel-basiertes System zum Aggregieren und Korrelieren von IDS-Meldungen. In diesem Zusammenhang ist eine wichtige Aufgabe, mehrere zusammengehörige Meldungen zu einem Alarm zusammenzufassen und die Anzahl der Meldungen zu reduzieren. Beispielsweise können viele Passwort-Rateversuche zu einem Alarm zusammengefasst werden. Eine weitere Aufgabe ist die Reduzierung der IDS-Fehlalarme. Um die Übersichtlichkeit der IDS-Meldungen zu erhalten, werden im GIDS nur korrelierte Alarme angezeigt. Die Aufgabe der Korrelation und Aggration im GIDS übernimmt der Prelude-Correlator, der speziell für das GIDS erweitert wurde.

B.2 Übersicht über die Sensorik

Snort

Snort ist ein Netzwerk-basiertes IDS, das in der Default-Konfiguration einen sehr umfangreichen Satz an Signaturen bietet. Signaturen sind sowohl für Angriffe auf Server als auch auf Clients wie beispielsweise Webbrowser vorhanden. Der Einsatzzweck von Snort ist die Erkennung automatisierter Angriffe, die insbesondere die Klasse der Internet-Würmer umfasst. Weiterhin bietet Snort eine gute Erkennung von Angriffen auf die typischen Serveranwendungen wie beispielsweise Webserver und Datenbankserver.

Prelude-LML

Prelude-LML ist ein lokaler IDS-Sensor, dessen Regelsatz die Überwachung einer Vielzahl von unterschiedlichen Log-Formaten bietet. Der Einsatz im GIDS ist die Erkennung von Portscans und Brute-Force-Angriffen auf SSH Server.

OSSEC

Analog zu Prelude-LML ist OSSEC ein lokaler IDS-Sensor. Neben der Überwachung von Log-Dateien kann OSSEC per kryptografischen Hashes Änderungen bei Systemdateien feststellen. Weiterhin bietet OSSEC eine gute heuristische Erkennung der Aktivitäten von Rootkits. Dies wird durch einen Vergleich der nativen API des Betriebssystem mit anderen Methoden realisiert. Ziel ist es, zu erkennen, ob ein Rootkit die API des Betriebssystem so manipuliert hat, dass beispielsweise Dateien oder Prozesse des Angreifer verborgen werden. Dazu vergleicht OSSEC zum Beispiel die Liste der Dateien in einem Verzeichnis, die auf der einen Seite durch die Betriebsystem API und auf der anderen Seite mit einem eigenen Dateisystem-Treiber erstellt wurde. Abweichungen weisen, wie bereits vorher beschrieben, auf die Aktivitäten eines Rootkits hin.

B.3 Installation und Konfiguration des `gids-client`

1. Installieren und konfigurieren Sie die Pakete `openvpn`, `mysql-server`, `libprelude-dev` und `prelude-manager`, die für die weitere Installation vorausgesetzt werden. Geben Sie dazu folgenden Befehl ein:

   ```
   apt-get install openvpn mysql-server libprelude-dev
   prelude-manager
   ```

   Hinweise:

   (a) Konfigurieren Sie ebenfalls die Datenbank für den `prelude-manager` mit Hilfe von `dbconfig-common`, wählen Sie als Datenbank `mysql` aus und geben Sie wiederum das zuvor festgelegte Passwort an (es folgen drei Abfragen).

(b) Beachten Sie, dass der `prelude-manager` nach der Installation nicht erfoglreich startet, was an dieser Stelle keinen Fehler darstellt. Die Ursache und Lösung ist nachfolgend beschrieben.

2. Prüfen Sie, ob der `prelude-manager` erfolgreich gestartet wurde, indem Sie in der Prozessliste Ihres Systems nach einem Prozess mit dem Namen `prelude-manager` suchen. Sollten Sie keinen solchen Prozess finden, gehen Sie wie folgt vor:

   (a) Editieren Sie die Datei `/etc/default/prelude-manager` und setzen Sie den Eintrag der Variablen `RUN` auf „`yes`“.

   (b) Starten Sie den `prelude-manager` neu. Geben Sie dazu folgenden Befehl ein:

   ```
   /etc/init.d/prelude-manager start
   ```

   (c) Prüfen Sie, ob der `prelude-manager` nun erfolgreich gestartet wurde, indem Sie in der Prozessliste Ihres Systems nach einem Prozess mit dem Namen `prelude-manager` suchen. Beachten Sie, dass das init-Skript des `prelude-manager` erfahrungsgemäß in manchen Situationen einen falschen Rückgabewert liefert, wobei der Start des Programms erfolgreich durchgeführt werden konnte!

3. Wechseln Sie in das Verzeichnis `/root`. Installieren Sie `emcast` und das Paket `gids-client`. Geben Sie dazu folgenden Befehl ein:

   ```
   dpkg -i libglib1.2ldbl_1.2.10-19_amd64.deb
   emcast_0.3.2-6_amd64.deb gids-client-0.2.deb
   ```

4. Installieren Sie nun Ihr Zertifikat und privaten Schlüssel von `openvpn` für den `gids-client`. Kopieren Sie diese nach `/etc/openvpn/gids-client/client.[crt|key]`.
   Beachten Sie, dass Sie die Rechte für den privaten Schlüssel aus Gründen der Sicherheit sehr restriktiv vergeben müssen. Geben Sie dazu folgenden Befehl ein:

   ```
   chmod 600 /etc/openvpn/gids-client/client.key
   ```

5. Passen Sie den Eintrag `gidsRessource` in der Konfigurationsdatei `/etc/ gids/gids-export.conf` an. Tragen Sie hier das Ihrer D-Grid-Site vergebene Kürzel ein, so wie es auch im GRRS zu finden ist. Diese Angabe ist wichtig, für die Zugriffskontrolle im GIDS-Portal. Es sind nur die Meldungen für Ressourcen sichtbar, auf die Sie Zugriff haben. Des Weiteren kann konfiguriert werden, welche Datenfelder anonymisiert bzw. gefiltert werden sollen. Die Default-Einstellungen sind so gewählt, dass keine Daten exportiert werden, die entweder Bezug zu einer natürlichen Person, oder als sicherheitskritisch eingestuft wurden.

6. Registrieren Sie die Komponente `gids-export` beim `prelude-manager`. In dem nachfolgenden Beispiel wird davon ausgegangen, der der Prelude-Manager auf dem glreichen System läuft. Ist dies nicht der Fall, muss die IP-Adresse des Prelude-Manager gewählt werden. Weiterhin wird davon ausgegangen, dass der GIDS-Export als Benutzer root läuft. Geben Sie dazu folgenden Befehl ein:

   ```
   prelude-admin register "gids-export" "idmef:rw" 127.0.0.1 --uid
   0 --gid 0
   ```

   (a) Öffnen Sie eine zweite Eingabeaufforderung (*Shell*) auf Ihrem System und starten Sie die Registrierung der neuen Komponente. Geben Sie dazu folgenden Befehl ein:

   ```
   prelude-admin registration-server prelude-manager
   ```

   (b) Kopieren Sie das angezeigte *one shot password* aus der Eingabeaufforderung.

   (c) Geben Sie das kopierte Passwort zweimal in Ihrer ersten Eingabeaufforderung ein.

   (d) Autorisieren Sie in der zweiten Eingabeaufforderung abschließend die Registrierung der neuen Komponente durch die Eingabe „`y`“.

7. Starten Sie den `gids-client`. Geben Sie dazu folgenden Befehl ein:

```
/etc/init.d/gids-client start
```

Hinweis: Der Start des `gids-client` kann einige Sekunden dauern.

B.4 Installation eines Test-Sensors

Nun da Sie die Installation und Konfiguration des `gids-client` angeschlossen haben, haben Sie die Möglichkeit einen Test-Sensor zu installieren, der künstlich erzeugte Nachrichten auf den GIDS-Bus legen kann. Dieser Sensor dient Ihnen zu Testzwecken des Daten-Exports. Zur Installation des Test-Sensors gehen Sie wie folgt vor:

1. Installieren Sie das Paket `python-prelude`. Dieses Paket wird zur Ausführung des Test-Sensors benötigt. Geben Sie dazu folgenden Befehl ein:

```
apt-get install python-prelude
```

2. Registrieren Sie den Test-Sensor beim `prelude-manager`. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin register "CorrelationSensor" "idmef:w" 127.0.0.1
--uid 0 --gid 0
```

   (a) Öffnen Sie eine zweite Eingabeaufforderung (*Shell*) auf Ihrem System und starten Sie die Registrierung der neuen Komponente. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin registration-server prelude-manager
```

   (b) Kopieren Sie das angezeigte *one shot password* aus der Eingabeaufforderung.

   (c) Geben Sie das kopierte Passwort zweimal in Ihrer ersten Eingabeaufforderung ein.

   (d) Autorisieren Sie in der zweiten Eingabeaufforderung abschließend die Registrierung der neuen Komponente durch die Eingabe „`y`“.

3. Um eine Testnachricht auf den GIDS-Bus zu legen, verwenden Sie das Programm *idmef_test.py*. Geben Sie dazu folgenden Befehl ein:

```
/root/idmef_test.py
```

B.5 Installation von `prelude-lml` als Sensor

Zur Installation von `prelude-lml` als Sensor gehen Sie bitte wie folgt vor:

1. Installieren Sie das Paket `prelude-lml`. Geben Sie dazu folgenden Befehl ein:

```
apt-get install prelude-lml
```

   Hinweis: Vermutlich erhalten Sie bei der Installation eine Fehlermeldung. Diese Fehlermeldung wird beim Versuch `prelude-lml` zu starten erzeugt, weil dieser Sensor noch nicht am `prelude-manager` registriert ist.

2. Registrieren Sie die Komponente `prelude-lml` beim `prelude-manager`. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin register "prelude-lml" "idmef:w" 127.0.0.1 --uid
prelude
```

(a) Öffnen Sie eine zweite Eingabeaufforderung (*Shell*) auf Ihrem System und starten Sie die Registrierung der neuen Komponente. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin registration-server prelude-manager
```

(b) Kopieren Sie das angezeigte *one shot password* aus der Eingabeaufforderung.

(c) Geben Sie das kopierte Passwort zweimal in Ihrer ersten Eingabeaufforderung ein.

(d) Autorisieren Sie in der zweiten Eingabeaufforderung abschließend die Registrierung der neuen Komponente durch die Eingabe „`y`“.

3. Schließen Sie die Installation von `prelude-lml` ab. Geben Sie dazu folgenden Befehl ein:

```
apt-get install prelude-lml
```

4. Prüfen Sie, ob `prelude-lml` nun erfolgreich gestartet wurde, indem Sie in der Prozessliste Ihres Systems nach einem Prozess mit dem Namen `prelude-lml` suchen. Sollte `prelude-lml` nicht gestartet worden sein, versuchen Sie `prelude-lml` neu zu starten. Geben Sie dazu folgenden Befehl ein:

```
/etc/init.d/prelude-lml restart
```

B.6 Installation von `snort` als Sensor

Zur Installation von `snort` als Sensor gehen Sie bitte wie folgt vor:

1. Installieren Sie das Paket `snort`. Geben Sie dazu folgenden Befehl ein:

```
apt-get install snort
```

Befolgen Sie bitte die Anweisungen des Installationsmenüs.

2. Editieren Sie die Datei `/etc/snort/snort.conf`. Aktivieren Sie in der Sektion „*prelude*“ die Anbindung an den `prelude-manager` durch hinzufügen folgender Zeile:
`output alert_prelude: profile=snort`

3. Per Default wird der vollständige Regelsatz installiert. Dieser Regelsatz enthält viele Regeln, die „False-Positives“ erzeugen können (beispielsweise ICMP-Pings). Im Portal werden allerdings nur korrelierte IDS-Meldungen angezeigt. Deshalb ist noch eine Stufe vorhanden, die „False-Positives“ unterdrückt. Je nach dem Einsatzzweck des Systems können die Regeln angepasst werden. Es werden aber nur korrelierte IDS-Meldungen angezeigt. Zum Beispiel wird vom GIDS-Projekt ein Regelsatz zur Verfügung gestellt, der speziell an Grid-Services angepasst ist. Die Installation erfolgt, indem die Datei mit den Regeln in das Verzeichnis mit den Snort-Regeln kopiert wird und in der Konfiguration abtiviert wird: `cp ...`
`aktivieren..`

4. Registrieren Sie die Komponente `snort` beim `prelude-manager`. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin register "snort" "idmef:w" 127.0.0.1 --uid snort
```

(a) Öffnen Sie eine zweite Eingabeaufforderung (*Shell*) auf Ihrem System und starten Sie die Registrierung der neuen Komponente. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin registration-server prelude-manager
```

(b) Kopieren Sie das angezeigte *one shot password* aus der Eingabeaufforderung.

(c) Geben Sie das kopierte Passwort zweimal in Ihrer ersten Eingabeaufforderung ein.

(d) Autorisieren Sie in der zweiten Eingabeaufforderung abschließend die Registrierung der neuen Komponente durch die Eingabe „y“.

5. Starten Sie `snort` neu. Geben Sie dazu folgenden Befehl ein:

```
/etc/init.d/snort restart
```

B.7 Installation des `prelude-correlator` als Sensor

Wie bereits oben geschrieben, ist der Prelude-Correlator eine zentrale Instanz zur Korrelation und Aggregation von IDS-Meldungen. Als wichtige Aufgabe dient dieser dazu, das GIDS robuster gegenüber Fehlalarmen zu machen. Aus diesem Grund werden im Portal ausschließlich korrelierte bzw. agregierte Alarme angezeigt. Der Correlator selbst ist in the Programmiersprache Python geschrieben worden und verbindet sich als Prelude-Sensor zu dem Prelude-Manager. Falls die Prelude-Manager in einer administrativen Domäne hierarchisch angeordnet sind, ist es deshalb wichtig, dass der Correlator sich zu dem zentralen Manager verbindet. Dort verarbeitet dieser alle IDS-Meldungen und erzeugt Korrelations-Alarme im Format IDMEF als Ausgabe.

Zur Installation von `prelude-correlators` als Sensor gehen Sie bitte wie folgt vor:

1. Installieren Sie das Paket `python-prelude`. Geben Sie dazu folgenden Befehl ein:

```
apt-get install python-prelude
```

2. Das Paket liegt als Tar-Archiv vor und wird mit Geben Sie dazu folgenden Befehl ein:

```
tar -xzvf prelude-gids-correlator-1.0.0.tar.gz
```

3. Die Installation der allgemeinen Regeln erfolgt im Verzeichnis *prelude-correlator-1.0.0* durch Geben Sie dazu folgenden Befehl ein:

```
python setup.py install
```

4. Danach werden die GIDS-Regeln mit dem obigen Befehl im Verzeichnis *PreludeCorrelator/gids-plugin/* installiert.

5. Zur Konfiguration kopieren Sie beispielsweise die Datei `prelude-correlator.conf` aus dem Verzeichnis `gids-plugin` nach `/etc/gids`. In der Konfigurationsdatei können die verschiedenen Plug-ins des Correlators aktiviert oder deaktiviert werden.

6. Registrieren Sie die Komponente `snort` beim `prelude-manager`. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin register "prelude-correlatior" "idmef:rw"
127.0.0.1 --uid 0 --gid 0
```

(a) Öffnen Sie eine zweite Eingabeaufforderung (*Shell*) auf Ihrem System und starten Sie die Registrierung der neuen Komponente. Geben Sie dazu folgenden Befehl ein:

```
prelude-admin registration-server prelude-manager
```

(b) Kopieren Sie das angezeigte *one shot password* aus der Eingabeaufforderung.

(c) Geben Sie das kopierte Passwort zweimal in Ihrer ersten Eingabeaufforderung ein.

(d) Autorisieren Sie in der zweiten Eingabeaufforderung abschließend die Registrierung der neuen Komponente durch die Eingabe „y“.

7. Der Start erfolgt über den Aufruf `prelude-correlator`: Geben Sie dazu folgenden Befehl ein:

```
prelude-correlator -c /etc/gids/prelude-correlator.conf
```

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Shirey, R. (Hrsg.): *Internet Security Glossary.* Mai 2000. – URL `http://www.ietf.org/rfc/rfc2828.txt`

[2] CAIDA – Cooperative Association for Internet Data Analysis: *The Spread of the Code-Red Worm (CRv2).* April 2012. – URL `http://www.caida.org/research/security/code-red/coderedv2_analysis.xml`

[3] gentschen Felde, N.: Einsatz der graphbasierten Meldungsstrukturanalyse in domänenübergreifenden Meta-IDS. In: *Lecture Notes in Informatics — Informatik 2005, Informatik LIVE!* Bonn, Germany : Gesellschaft für Informatik, September 2005 (Band 2 P–68), S. 653–657

[4] gentschen Felde, N. ; Jahnke, M. ; Martini, P. ; Tölle, J.: Impact of Sanitized Message Flows in a Cooperative Intrusion Warning System. In: *Proceedings of the 25th Military Communications Conference (MILCOM 2006)* Bd. 2006. Washington, DC, USA, Oktober 2006, S. 1–7

[5] Hommel, Wolfgang ; gentschen Felde, Nils ; von Eye, Felix ; Kohlrausch, Jan ; Szongott, Christian: Architekturkonzept fur ein Grid-basiertes IDS / D-Grid. URL `http://www.grid-ids.de/documents/GIDS_MS16-1.pdf`, Oktober 2010. – Meilensteinbericht

[6] Hommel, Wolfgang ; gentschen Felde, Nils ; von Eye, Felix ; Kohlrausch, Jan ; Szongott, Christian: Datenschutzmodell für ein Grid-basiertes IDS / D-Grid. URL `http://www.grid-ids.de/documents/GIDS_MS13.pdf`, Juli 2010. – Meilensteinbericht

[7] Hommel, Wolfgang ; gentschen Felde, Nils ; von Eye, Felix ; Kohlrausch, Jan ; Szongott, Christian: Grobskizze einer Architektur / D-Grid. URL `http://www.grid-ids.de/documents/GIDS_MS10.pdf`, April 2010. – Meilensteinbericht

[8] Hommel, Wolfgang ; gentschen Felde, Nils ; von Eye, Felix ; Kohlrausch, Jan ; Szongott, Christian: Prototypische Implementierung / D-Grid. URL `http://www.grid-ids.de/documents/GIDS_MS28.pdf`, Februar 2012. – Meilensteinbericht

[9] Reiser, Helmut ; gentschen Felde, Nils ; von Eye, Felix ; Kohlrausch, Jan ; Szongott, Christian: Anforderungs- und Kritierienkatalog (MS 6) / D-Grid. URL `http://www.grid-ids.de/documents/GIDS_MS6.pdf`, Januar 2010. – Meilensteinbericht

[10] Schmidt, Harald: *Simulation und Erkennung der Ausbreitungsstruktur von Würmern*, Universität Bonn, Diplomarbeit, September 2002

[11] Symantec: *Symantec Security Response: „FreeBSD.Scalper.Worm“.* Juli 2002. – URL `http://securityresponse.symantec.com/avcenter/venc/data/freebsd.scalper.worm.html`

[12] Symantec: *Symantec Security Response: „CodeRed II“.* Februar 2007. – URL `http://www.symantec.com/security_response/writeup.jsp?docid=2001-080421-3353-99`

[13] Symantec: *Symantec Security Response: „CodeRed Worm“.* Februar 2007. – URL `http://www.sarc.com/avcenter/venc/data/codered.worm.html`