

Konzeption und Implementierung einer Middleware zur Manipulation von Netzerkdiensten und -verbindungen in einem hochverfügbaren Gateway-Cluster

Alexander von Gernler

Diplomarbeit im Fach Informatik
Februar – September 2005

Inhaltsübersicht

1 Problemstellung

- Kurzvorstellung GeNUA
- Beschreibung GeNUGate
- Ausgangssituation Relays
- Wunschvorstellung

2 Planungsphase

- Analyse
- Aufstellung eines Projekts
- Entwurf
- Related Work

3 Realisierungsphase

- commd
- Nachrichtenformat
- IMMSG.pm
- comm.pl

4 Fazit und Ausblick

- Vorteile
- Nachteile
- TODO

Inhaltsübersicht

1 Problemstellung

- Kurzvorstellung GeNUA
- Beschreibung GeNUGate
- Ausgangssituation Relays
- Wunschvorstellung

2 Planungsphase

- Analyse
- Aufstellung eines Projekts
- Entwurf
- Related Work

3 Realisierungsphase

- commd
- Nachrichtenformat
- IMMSG.pm
- comm.pl

4 Fazit und Ausblick

- Vorteile
- Nachteile
- TODO

Wer oder was ist GeNUA?

- Mittelständisches IT-Unternehmen mit Sitz in Kirchheim bei München
- ca. 70 Mitarbeiter
- spezialisiert auf Absicherung von Netzwerken und IT-Security
- Referenzkunden z. B. Deutscher Bundestag, Innenministerium, MAN, Burda Media, RTL II
- Produktreihen
 - GeNUGate – zweistufige Firewall, clusterfähig
 - GeNUBox – Krypto- und VPN Box
 - GeNUDetect – Intrusion Detection System
 - GeNULink – Link Balancer



Wer oder was ist GeNUA?

- Mittelständisches IT-Unternehmen mit Sitz in Kirchheim bei München
- ca. 70 Mitarbeiter
- spezialisiert auf Absicherung von Netzwerken und IT-Security
- Referenzkunden z. B. Deutscher Bundestag, Innenministerium, MAN, Burda Media, RTL II
- Produktreihen
 - GeNUGate – zweistufige Firewall, clusterfähig
 - GeNUBox – Krypto- und VPN Box
 - GeNUDetect – Intrusion Detection System
 - GeNULink – Link Balancer



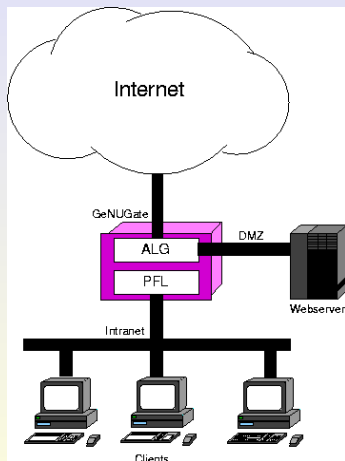
Wer oder was ist GeNUA?

- Mittelständisches IT-Unternehmen mit Sitz in Kirchheim bei München
- ca. 70 Mitarbeiter
- spezialisiert auf Absicherung von Netzwerken und IT-Security
- Referenzkunden z. B. Deutscher Bundestag, Innenministerium, MAN, Burda Media, RTL II
- Produktreihen
 - GeNUGate – zweistufige Firewall, clusterfähig
 - GeNUBox – Krypto- und VPN Box
 - GeNUDetect – Intrusion Detection System
 - GeNULink – Link Balancer



GeNUGate Kurzübersicht

- Firewall auf zwei unabhängigen *physikalischen* Rechnern
 - 1 ALG (Application Level Gateway)
 - 2 PFL (Paketfilter)
- Einsatz einzeln oder im HA-Cluster. Bisher $n \leq 8$.
- Bei Cluster-Einsatz eigenes HA-Zwischennetz und OSPF-Router davor
- BSI-sicherheitszertifiziert
 - ITSEC E3/hoch
 - **Common Criteria EAL 4+**
- Relay-Prozesse auf ALG reichen Netzwerkpakete durch



Quelle: GeNUA

Was ist ein Relay?

- 1 Lauscht auf je einem bestimmten Port nach Netzwerkpaketen
- 2 Nimmt diese entgegen, prüft Validität
- 3 Entfernt evtl. unerwünschte Inhalte (bei HTML z. B. JavaScript, ActiveX, Cookies etc.)
- 4 Nimmt z. B. Prüfung auf Viren vor (bei Mails und Downloads)
- 5 Reicht Paket weiter, wenn alles in Ordnung

Grundsätzliche Regeln

- Kein Routing auf OSI-Schicht 3 im ALG
- Alle Pakete müssen ein Relay passieren

Was ist ein Relay?

- 1 Lauscht auf je einem bestimmten Port nach Netzwerkpaketen
- 2 Nimmt diese entgegen, prüft Validität
- 3 Entfernt evtl. unerwünschte Inhalte (bei HTML z. B. JavaScript, ActiveX, Cookies etc.)
- 4 Nimmt z. B. Prüfung auf Viren vor (bei Mails und Downloads)
- 5 Reicht Paket weiter, wenn alles in Ordnung

Grundsätzliche Regeln

- Kein Routing auf OSI-Schicht 3 im ALG
- Alle Pakete müssen ein Relay passieren

Status quo

- Relays lesen ihre Konfiguration beim Start
- Während des Programmlaufs praktisch keine Kommunikation mit den Relays möglich
- Neue Konfiguration erfordert Neustart des Prozesses (mit allen Konsequenzen: Verbindungsverlust, Unerreichbarkeit)
- Debugging für Entwickler und Supporter schwierig
- Manipulation des Zustands zur Laufzeit nicht möglich

Also Erstellung einer Nachrichtenschnittstelle zu den Relays nötig!

Ziele und Anforderungen von GeNUA

- Nachrichtenschicht, mit der der Admin von außen die Relays ansteuern kann
- Statusabfrage, Debugging und Termination einzelner Verbindungen und Prozessen per Benutzeroberfläche
- Firewall ist hochverfügbar, Schnittstelle muss das auch sein
- OpenSource Paradigma in der Firma (OpenBSD, Perl)
- BSI-Rezertifizierung angestrebt: Sicherheitsschwerpunkt
- Ressourcenschonende, erweiterbare, gut dokumentierte Implementierung

Ziele und Anforderungen von GeNUA

- Nachrichtenschicht, mit der der Admin von außen die Relays ansteuern kann
- Statusabfrage, Debugging und Termination einzelner Verbindungen und Prozessen per Benutzeroberfläche
- Firewall ist hochverfügbar, Schnittstelle muss das auch sein
- OpenSource Paradigma in der Firma (OpenBSD, Perl)
- BSI-Rezertifizierung angestrebt: Sicherheitsschwerpunkt
- Ressourcenschonende, erweiterbare, gut dokumentierte Implementierung

Ziele und Anforderungen von GeNUA

- Nachrichtenschicht, mit der der Admin von außen die Relays ansteuern kann
- Statusabfrage, Debugging und Termination einzelner Verbindungen und Prozessen per Benutzeroberfläche
- Firewall ist hochverfügbar, Schnittstelle muss das auch sein
- OpenSource Paradigma in der Firma (OpenBSD, Perl)
- BSI-Rezertifizierung angestrebt: Sicherheitsschwerpunkt
- Ressourcenschonende, erweiterbare, gut dokumentierte Implementierung

Ziele und Anforderungen von GeNUA

- Nachrichtenschicht, mit der der Admin von außen die Relays ansteuern kann
- Statusabfrage, Debugging und Termination einzelner Verbindungen und Prozessen per Benutzeroberfläche
- Firewall ist hochverfügbar, Schnittstelle muss das auch sein
- OpenSource Paradigma in der Firma (OpenBSD, Perl)
- BSI-Rezertifizierung angestrebt: Sicherheitsschwerpunkt
- Ressourcenschonende, erweiterbare, gut dokumentierte Implementierung

Ziele und Anforderungen von GeNUA

- Nachrichtenschicht, mit der der Admin von außen die Relays ansteuern kann
- Statusabfrage, Debugging und Termination einzelner Verbindungen und Prozessen per Benutzeroberfläche
- Firewall ist hochverfügbar, Schnittstelle muss das auch sein
- OpenSource Paradigma in der Firma (OpenBSD, Perl)
- BSI-Rezertifizierung angestrebt: Sicherheitsschwerpunkt
- Ressourcenschonende, erweiterbare, gut dokumentierte Implementierung

Ziele und Anforderungen von GeNUA

- Nachrichtenschicht, mit der der Admin von außen die Relays ansteuern kann
- Statusabfrage, Debugging und Termination einzelner Verbindungen und Prozessen per Benutzeroberfläche
- Firewall ist hochverfügbar, Schnittstelle muss das auch sein
- OpenSource Paradigma in der Firma (OpenBSD, Perl)
- BSI-Rezertifizierung angestrebt: Sicherheitsschwerpunkt
- Ressourcenschonende, erweiterbare, gut dokumentierte Implementierung

Inhaltsübersicht

- 1 Problemstellung
 - Kurzvorstellung GeNUA
 - Beschreibung GeNUGate
 - Ausgangssituation Relays
 - Wunschvorstellung
- 2 **Planungsphase**
 - Analyse
 - Aufstellung eines Projekts
 - Entwurf
 - Related Work
- 3 Realisierungsphase
 - commd
 - Nachrichtenformat
 - IMMSG.pm
 - comm.pl
- 4 Fazit und Ausblick
 - Vorteile
 - Nachteile
 - TODO

Wissenschaftliche Formulierung (1)

Praktische Sicht: GeNUA

„Wir brauchen eine Kommunikationsschicht!“

- Praktische Implementierungen ungeeignet
 - Microsoft .net proprietär bzw. patentgefährdet
 - OMG CORBA zu mächtig für Zertifizierung
 - Sun RPC nur synchroner Aufruf, Blockieren aber inakzeptabel
 - Java RMI fällt aus wegen Festlegung auf Perl
 - NCSA VMI keine freie Implementation auffindbar, außerdem Schwerpunkt High Performance Cluster
 - MPI zu sehr auf HP-Cluster ausgelegt, außerdem Spezifikation noch nicht 100% implementiert

Wissenschaftliche Formulierung (1)

Praktische Sicht: GeNUA

„Wir brauchen eine Kommunikationsschicht!“

- Praktische Implementierungen ungeeignet
 - Microsoft .net proprietär bzw. patentgefährdet
 - OMG CORBA zu mächtig für Zertifizierung
 - Sun RPC nur synchroner Aufruf, Blockieren aber inakzeptabel
 - Java RMI fällt aus wegen Festlegung auf Perl
 - NCSA VMI keine freie Implementation auffindbar, außerdem Schwerpunkt High Performance Cluster
 - MPI zu sehr auf HP-Cluster ausgelegt, außerdem Spezifikation noch nicht 100% implementiert

Wissenschaftliche Formulierung (1)

Praktische Sicht: GeNUA

„Wir brauchen eine Kommunikationsschicht!“

- Praktische Implementierungen ungeeignet
 - Microsoft .net proprietär bzw. patentgefährdet
 - OMG CORBA zu mächtig für Zertifizierung
 - Sun RPC nur synchroner Aufruf, Blockieren aber inakzeptabel
 - Java RMI fällt aus wegen Festlegung auf Perl
 - NCSA VMI keine freie Implementation auffindbar, außerdem Schwerpunkt High Performance Cluster
 - MPI zu sehr auf HP-Cluster ausgelegt, außerdem Spezifikation noch nicht 100% implementiert

Wissenschaftliche Formulierung (1)

Praktische Sicht: GeNUA

„Wir brauchen eine Kommunikationsschicht!“

- Praktische Implementierungen ungeeignet
 - Microsoft .net proprietär bzw. patentgefährdet
 - OMG CORBA zu mächtig für Zertifizierung
 - Sun RPC nur synchroner Aufruf, Blockieren aber inakzeptabel
 - Java RMI fällt aus wegen Festlegung auf Perl
 - NCSA VMI keine freie Implementation auffindbar, außerdem Schwerpunkt High Performance Cluster
 - MPI zu sehr auf HP-Cluster ausgelegt, außerdem Spezifikation noch nicht 100% implementiert

Wissenschaftliche Formulierung (1)

Praktische Sicht: GeNUA

„Wir brauchen eine Kommunikationsschicht!“

- Praktische Implementierungen ungeeignet
 - Microsoft .net proprietär bzw. patentgefährdet
 - OMG CORBA zu mächtig für Zertifizierung
 - Sun RPC nur synchroner Aufruf, Blockieren aber inakzeptabel
 - Java RMI fällt aus wegen Festlegung auf Perl
 - NCSA VMI keine freie Implementation auffindbar, außerdem Schwerpunkt High Performance Cluster
 - MPI zu sehr auf HP-Cluster ausgelegt, außerdem Spezifikation noch nicht 100% implementiert

Wissenschaftliche Formulierung (1)

Praktische Sicht: GeNUA

„Wir brauchen eine Kommunikationsschicht!“

- Praktische Implementierungen ungeeignet
 - Microsoft .net proprietär bzw. patentgefährdet
 - OMG CORBA zu mächtig für Zertifizierung
 - Sun RPC nur synchroner Aufruf, Blockieren aber inakzeptabel
 - Java RMI fällt aus wegen Festlegung auf Perl
 - NCSA VMI keine freie Implementation auffindbar, außerdem Schwerpunkt High Performance Cluster
 - MPI zu sehr auf HP-Cluster ausgelegt, außerdem Spezifikation noch nicht 100% implementiert

Wissenschaftliche Formulierung (2)

Akademische Sicht: Diplomarbeit

Kernproblem: Entfernter Methodenaufruf

- Wissenschaftlich erschöpfend behandelt
- Theoretische Konzepte alle vorhanden
 - Verteilte Systeme
 - Middleware
 - Entwurfsmuster
- Aufgabe: Suche bzw. Erstellung einer *lean implementation*
- Herausforderung HA-Cluster, Fluktuation von Knoten
- Mangels Alternativen Neuimplementation
 - Lizenzproblematik für GeNUA gelöst
 - Volle Kontrolle über Code
 - Kein unnötiger Ballast

Wissenschaftliche Formulierung (2)

Akademische Sicht: Diplomarbeit

Kernproblem: Entfernter Methodenaufruf

- Wissenschaftlich erschöpfend behandelt
- Theoretische Konzepte alle vorhanden
 - Verteilte Systeme
 - Middleware
 - Entwurfsmuster
- Aufgabe: Suche bzw. Erstellung einer *lean implementation*
- Herausforderung HA-Cluster, Fluktuation von Knoten
- Mangels Alternativen Neuimplementation
 - Lizenzproblematik für GeNUA gelöst
 - Volle Kontrolle über Code
 - Kein unnötiger Ballast

Systemidee (OEP)

„Auf dem Produkt GeNUGate soll die clusterorientierte Ansteuerung der Relay-Prozesse durch Neuentwicklung einer Schnittstelle ermöglicht werden.

Diese Kommunikationsschicht soll ein entferntes Aufrufen von Methoden auf den Relays ungeachtet ihrer Lage oder Funktion im Cluster gestatten. Dazu gehört die Referenzimplementation einer Basismenge aufrufbarer Funktionen und das Vorsehen zukünftiger einfacher Erweiterbarkeit. Außerdem ist eine textorientierte Bedienoberfläche als *proof of concept* zu entwerfen und zu implementieren, sowie ein Konzept für die Darstellung in einer graphischen Oberfläche zu erarbeiten.

Eine Neuimplementation bisher nicht existierender Funktionalität in den Relays, sowie eine vollständige graphische Oberfläche sind nicht Bestandteil dieser Diplomarbeit.“

Zeitplan

Feb-Mär Einarbeitung ins Thema, Literaturrecherche

Mär-Apr Entwurf und externe Implementation des `cmd`

Apr-Mai Schreiben und Testen des „Klebers“: `IMSG.pm`

Mai-Jun Integration des externen Codes in die GeNUA Codebasis

Mai-Jun Implementation der Kommandozeileneingabe

Jul-Aug Ausblick auf GUI-Funktionalität

- Parallel hierzu Erstellung und Fortführung des DA-Skripts
- Regelmäßige Treffen mit JÜRGEN KLEINÖDER
- Betreuer in der Firma nur 2 Büros weiter

Zeitplan

Feb-Mär Einarbeitung ins Thema, Literaturrecherche

Mär-Apr Entwurf und externe Implementation des `cmd`

Apr-Mai Schreiben und Testen des „Klebers“: `IMSG.pm`

Mai-Jun Integration des externen Codes in die GeNUA Codebasis

Mai-Jun Implementation der Kommandozeileneingabe

Jul-Aug Ausblick auf GUI-Funktionalität

- Parallel hierzu Erstellung und Fortführung des DA-Skripts
- Regelmäßige Treffen mit JÜRGEN KLEINÖDER
- Betreuer in der Firma nur 2 Büros weiter

Zeitplan

Feb-Mär Einarbeitung ins Thema, Literaturrecherche

Mär-Apr Entwurf und externe Implementation des `cmd`

Apr-Mai Schreiben und Testen des „Klebers“: `IMSG.pm`

Mai-Jun Integration des externen Codes in die GeNUA Codebasis

Mai-Jun Implementation der Kommandozeileneingabe

Jul-Aug Ausblick auf GUI-Funktionalität

- Parallel hierzu Erstellung und Fortführung des DA-Skripts
- Regelmäßige Treffen mit JÜRGEN KLEINÖDER
- Betreuer in der Firma nur 2 Büros weiter

Zeitplan

Feb-Mär Einarbeitung ins Thema, Literaturrecherche

Mär-Apr Entwurf und externe Implementation des `cmd`

Apr-Mai Schreiben und Testen des „Klebers“: `IMSG.pm`

Mai-Jun Integration des externen Codes in die GeNUA Codebasis

Mai-Jun Implementation der Kommandozeileneingabe

Jul-Aug Ausblick auf GUI-Funktionalität

- Parallel hierzu Erstellung und Fortführung des DA-Skripts
- Regelmäßige Treffen mit JÜRGEN KLEINÖDER
- Betreuer in der Firma nur 2 Büros weiter

Zeitplan

Feb-Mär Einarbeitung ins Thema, Literaturrecherche

Mär-Apr Entwurf und externe Implementation des `cmd`

Apr-Mai Schreiben und Testen des „Klebers“: `IMSG.pm`

Mai-Jun Integration des externen Codes in die GeNUA Codebasis

Mai-Jun Implementation der Kommandozeileneingabe

Jul-Aug Ausblick auf GUI-Funktionalität

- Parallel hierzu Erstellung und Fortführung des DA-Skripts
- Regelmäßige Treffen mit JÜRGEN KLEINÖDER
- Betreuer in der Firma nur 2 Büros weiter

Zeitplan

Feb-Mär Einarbeitung ins Thema, Literaturrecherche

Mär-Apr Entwurf und externe Implementation des `cmd`

Apr-Mai Schreiben und Testen des „Klebers“: `IMSG.pm`

Mai-Jun Integration des externen Codes in die GeNUA Codebasis

Mai-Jun Implementation der Kommandozeileneingabe

Jul-Aug Ausblick auf GUI-Funktionalität

- Parallel hierzu Erstellung und Fortführung des DA-Skripts
- Regelmäßige Treffen mit JÜRGEN KLEINÖDER
- Betreuer in der Firma nur 2 Büros weiter

Idee der Kommunikationsschicht

- Realisierung von entferntem asynchronem Methodenaufruf
 - *call-by-value/result*
 - *exactly-once*
 - *Promises* (LISKOV et al.)
- Knotencontroller
 - *Beobachter* im Bezug auf Relays (GAMMA et al.)
 - *Reactor* im Bezug auf generierte Nachrichten (SCHMIDT)
 - *Guardian* im Bezug auf Knoten (LISKOV et al.)
- Anfrage des Controllers (*Polling*) und auch Senden von Benachrichtigungen (*Traps*) durch Relays
- *Fail-Stop* Semantik auf Knotenebene
- Automatischer Neustart defekter Relays vorgegeben
- Administrator verbindet sich auf beliebigen Knoten des Clusters, transparente Weiterleitung seiner Anfragen

Textorientierte Benutzeroberfläche

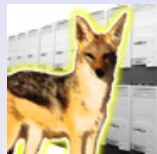
- Möglichkeiten zur *Selektion* von Relays aus der Gesamtmenge aller Relays aus einem Cluster, Spezialfälle
 - alle Relays
 - ein Relay
 - leere Menge
- Intention: SQL nachempfundene Selektionssprache
- Möglichkeit, auf selektierten Mengen *Operationen* auszuführen
- Menge der Operationen erweiterbar und nicht Stoff der DA
- Wissenschaftlicher Anspruch eher gering, daher schnellere proof-of-concept Implementation

Graphische Benutzeroberfläche

- War eigentlich eigenes DA-Thema bei GeNUA
- Daher nur Ausblick
- Implementation bei GeNUA starr vorgegeben (Web-GUI)
- GUI-Programmierung ohnehin eher „Handwerk“

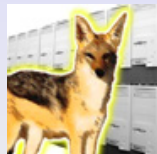
Jackal Software-DSM

- *DSM: Distributed Shared Memory*
- Projekt am Lehrstuhl 2: Behandlung der nicht-uniformen Speicherzugriffshierarchie in Clustern
- Erweiterung eines Java-Laufzeitsystems zum Transport von Nachrichten in einem Rechnerbündel
- Bearbeitung in Diplomarbeiten und Studienarbeiten
 - „Design und Implementation eines Kommunikationspaketes für Jackal“, Diplomarbeit von FRANK TRÖGER
 - „Erweiterung der Kommunikationsbibliothek Lizard“, Studienarbeit von BENJAMIN BIEBER
- Für uns nicht brauchbar, da im Zusammenhang mit Java und Ausrichtung auf High Performance



Jackal Software-DSM

- *DSM: Distributed Shared Memory*
- Projekt am Lehrstuhl 2: Behandlung der nicht-uniformen Speicherzugriffshierarchie in Clustern
- Erweiterung eines Java-Laufzeitsystems zum Transport von Nachrichten in einem Rechnerbündel
- Bearbeitung in Diplomarbeiten und Studienarbeiten
 - „Design und Implementation eines Kommunikationspaketes für Jackal“, Diplomarbeit von FRANK TRÖGER
 - „Erweiterung der Kommunikationsbibliothek Lizard“, Studienarbeit von BENJAMIN BIEBER
- Für uns nicht brauchbar, da im Zusammenhang mit Java und Ausrichtung auf High Performance



Netzmanagement

- SNMP: Simple Network Management Protocol
 - SNMPv1, SNMPv3 (RFC 3584)
 - RMON (RFC 2819)
- CMIP: Common Management Information Protocol
 - Vorschlag des OSI als Ersatz bzw. Verbesserung von SNMP
 - Nur in der Telekommunikation verbreitet, nicht in der IT
 - RFC 1189

Nachteile von SNMP

- Benötigt ASN.1 Parser (teuer, aufwendig, fehleranfällig)
- Kann Werte setzen/lesen, aber ungeeignet für entfernten Methodenaufruf

Netzmanagement

- SNMP: Simple Network Management Protocol
 - SNMPv1, SNMPv3 (RFC 3584)
 - RMON (RFC 2819)
- CMIP: Common Management Information Protocol
 - Vorschlag des OSI als Ersatz bzw. Verbesserung von SNMP
 - Nur in der Telekommunikation verbreitet, nicht in der IT
 - RFC 1189

Nachteile von SNMP

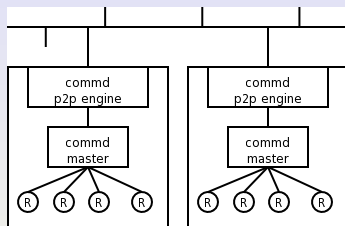
- Benötigt ASN.1 Parser (teuer, aufwendig, fehleranfällig)
- Kann Werte setzen/lesen, aber ungeeignet für entfernten Methodenaufruf

Inhaltsübersicht

- 1 Problemstellung
 - Kurzvorstellung GeNUA
 - Beschreibung GeNUGate
 - Ausgangssituation Relays
 - Wunschvorstellung
- 2 Planungsphase
 - Analyse
 - Aufstellung eines Projekts
 - Entwurf
 - Related Work
- 3 **Realisierungsphase**
 - **commd**
 - **Nachrichtenformat**
 - **IMSG.pm**
 - **comm.pl**
- 4 Fazit und Ausblick
 - Vorteile
 - Nachteile
 - TODO

Kommunikationsdienst commd

- Eine Instanz pro ALG im Cluster
- Kontrolle aller lokalen Relays
- Nachrichtenweitergabe im Cluster
- Privilegiensepariert, wegen Relay-Grundsatz: Kein Dienst lauscht *direkt* am Netz
- Implementation in C
 - Bereits laufender, freier Code verfügbar
 - Programmiersprache verschieden von der bei IMMSG Subsystem, wirft mehr Spezialsituationen auf, dadurch stabilerer Code
- Verwendung von `libevent` von NIELS PROVOS: Abstraktion des I/O-Multiplexing (`select(2)`, `poll(2)`, `kqueue(2)`)



Vorbilder aus der Freien Software

- OpenNTPD von HENNING BRAUER, *lean implementation* für `xntpd`



- Realisiert in C
 - OpenBSD KNF Styleguide `style(9)`
 - privilegiensepariert (vgl. Paper von PROVOS und FRIEDL)
 - Sehr gut dokumentiert
 - Eigener schlanker Nachrichtenmechanismus IMSG
- Viele Anleihen aus BSD-lizenziertem Code und Konzepten aus OpenNTPD: Das Rad nicht zweimal erfinden

On the wire – IMSG Paketformat

Definition der Struktur (C-Syntax):

```
struct imsg_hdr {
    u_int16_t      protocol;
    u_int16_t      type;
    u_int32_t      peerid;
    u_int32_t      pid;
    u_int16_t      len;
};
```

Packen einer IMSG in Perl:

```
my $imsg = pack("nnNNn", IMSG_PROTO_VERSION, $type,
    $ip, $$, $len) . $data);
```

Perl-Modul mit IMSG-Bibliothek

- Relay-Prozesse auf dem GeNUGate sind in Perl geschrieben
- gemeinsame Funktionen in GeNUA-eigener Bibliothek
Relay.pm
- Verbindung zur Kommunikationsschicht also Perl-Modul
- Schreiben von IMSG.pm (objektorientiert)
 - `new($sockname)`
 - `read_dispatcher()`
 - `write_dispatcher()`
 - `want_write()`
 - `getfd()`
 - `register_callback($type, $callback)`
 - `remove_callback($type)`
 - `compose($type, $ip, $data)`

Kommandozeilenoberfläche

- Verwendung von `Term::Shell`, bekannt z. B. aus Türschließsystem der Informatik gk (THOMAS GLANZMANN, MICHAEL GERNOTH)
- Modul existierte unter OpenBSD nicht: Eigenen Port gebaut und in-Tree gebracht
- Stellt Frontend für `IMSG.pm` dar.
- Abstrahiert alle Massnahmen, die für eine vernünftige Text-Shell nötig sind.
- Einfach zu bedienen, also schnell um Funktionalität erweiterbar:
 - `sub run_cmd()` Code für Befehl `cmd`
 - `sub help_cmd()` Hilfe dafür
 - `sub smry_cmd()` Kurzbeschreibung

Inhaltsübersicht

- 1 Problemstellung
 - Kurzvorstellung GeNUA
 - Beschreibung GeNUGate
 - Ausgangssituation Relays
 - Wunschvorstellung
- 2 Planungsphase
 - Analyse
 - Aufstellung eines Projekts
 - Entwurf
 - Related Work
- 3 Realisierungsphase
 - commd
 - Nachrichtenformat
 - IMMSG.pm
 - comm.pl
- 4 **Fazit und Ausblick**
 - Vorteile
 - Nachteile
 - TODO

Vorteile

- Schicht erfüllt die Anforderungen
 - Asynchroner entfernter Methodenaufruf
 - Spezielle, schlanke Lösung wegen Sicherheitszertifizierung
- Produkt ist weitgehend wiederverwendbar
 - `IMSG.pm` wird verwendet in
 - Relays (zu kontrollierende Elemente)
 - `comm.pl` (kontrollierendes Element)
 - `commd`
 - C-Code mit Syntactic Sugar (`style(9)`)
 - gut dokumentiert
 - paranoideste `-W` Optionen
 - sehr modular aufgebaut (`imsg.c`, `p2p.c`, `config.c`, ...)
 - Kommunikationsschicht auch in anderen Produkten von GeNUA einsetzbar

Nachteile

- Kein Internet-Standard, mangelnde Portabilität
 - Relativ wenig störend, da keine Interoperabilität mit anderen Produkten gewünscht.
- Keine Stumpfgeneratoren vorhanden, die Entwicklung erleichtern
 - Mißbrauch von `rpcgen` (vgl. `DUG SONGS xpw`) fällt wegen Perl aus
 - Neue Funktionen müssen „per Hand“ in Kommunikationsschicht gepflegt werden
 - Schicht zum Glück sehr leichtgewichtig, daher wenig Arbeit an wenigen definierten Stellen

Es gibt viel zu tun!

- Verbesserungen handwerklicher Art, die in DA keinen Platz fanden, weil zu zeitintensiv oder zu wenig wissenschaftlich.
 - Anpassungen aufs Produkt
 - IPv6 Support
 - Realisierung der Sicherheitsarchitektur (TCP MD5, IPsec, SSL?)
 - Evtl. Erweiterung und Anpassung für Einsatz in anderen Produkten
- Dennoch nötig
- Gegenstand meiner Weiterbeschäftigung bei GeNUA

Noch Fragen?



Kolophon

- 1 Folien erstellt mit \LaTeX , latex-beamer, make und CVS
- 2 Folien erhältlich unter <http://pestilenz.org/~grunk/vortraege/2005/da/da.pdf>
- 3 Quellcode der Folien auf Anfrage: `<grunk@openbsd.org>`