genua

Trafficanalyse und -modifikation mit IMP

Steffen Ullrich, genua mbH

Deutscher Perl-Workshop 2013, Berlin

Bundesministerium
für Bildung
und Forschung
genua

- Steffen Ullrich
- Perl-Entwickler seit 1996
  - cpan SULLR
  - github.com/noxxi
  - noxxi.de - Code, Talks..
- seit 2001 bei genua mbH
  - Entwicklung von Firewalls für Hochsicherheitsumgebungen
  - www.genua.de - Jobs, Abschlussarbeiten...
- seit 2011 Forschungsprojekt Padiofire
  - Web 2.0 aus Sicht von Perimeterfirewalls
  - zusammen mit Unis Cottbus, Erlangen, Innsbruck
  - gefördert vom BMBF

genua

- was machen Firewalls
- was ist IMP und wie funktioniert es
- existente Implementationen
- Schreiben eigener Trafficanalysen

genua

- Daten empfangen
- Daten analysieren
- Daten weiterreichen oder blockieren
- evtl. manipulieren, normalisieren

genua

- Loggen von Verbindungen, evtl nach SSL/SSH Bridging
- in SSL/SSH Tunneln erlaubte Protokolle einschränken
- Normalisieren von Web-Traffic
- Filtern mit URL-Blacklists
- Virenscanning
- Angriffserkennung
- Ads/Malvertising blocken
- DNS Filtern
- ...

genua

- Proxy, Application Level Gateway
- IDS
- Packet Captures
- ...

Trennung Datenquellen und Analyse

warum

- Entkoppeln Datenquellen von Analyse, damit...
- Re-Use von Analysen
- unabhängige Optimierung der Teile
- weniger Wartungsaufwand
- steigender Output/Qualität bei sinkendem Aufwand
- Konzentration auf die interessanten Sachen

genua

- einfach zu nutzen und implementieren
- jedoch flexibel genug
- streamingfähig
- performant
- stabil bzw. abwärtskompatibel

Trennung Datenquellen und Analyse

was heisst performant

- geringe Latenz
- hoher Durchsatz
- Wege
  - geringer Overhead
  - kein unnötiges Kopieren
  - kein unnötiges Analysieren

genua

- ICAP - Internet Content Adaption Protocol (2003)
  - primär HTTP, aber auch API Virenscanner
  - nur vollständige Anfragen/Antworten
  - Modifikation nur alles oder nichts
  - nicht tauglich
- OPES - Open Pluggable Edge Services (2005)
  - "ICAP/2.0"
  - streaming fähig
  - deutlich leistungsfähiger als ICAP
  - und deutlich komplexer (11 RFCs für Basis, HTTP, SMTP)
  - nirgendwo implementiert
- ???

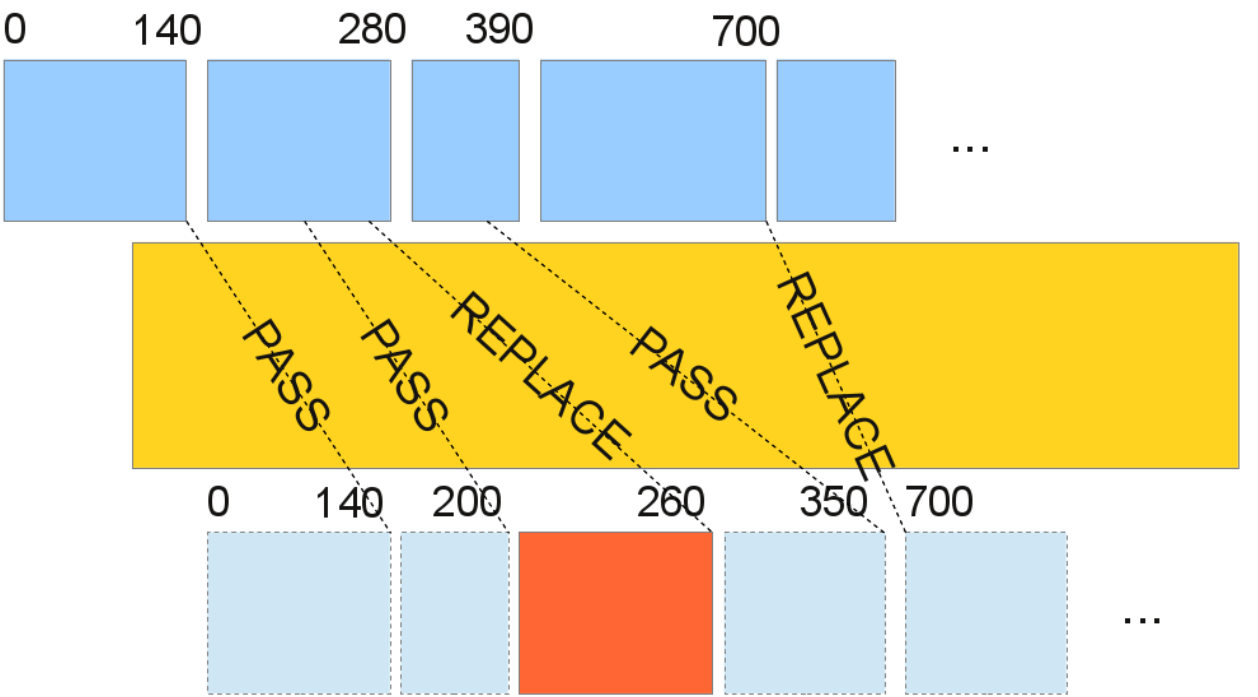
genua

- [I]nspection and [M]odification [P]rotocol
- Arbeit auf Datenchunks
- Kopie nur zur Analyse
  - kein Rückkopieren wenn unverändert
- asynchron, Feedback per Callback
  - DNS Anfragen o.ä blockieren System nicht
- Optimierungen zur Vermeidung von Analysen
- Optimierungen für geringe Latenz


ICAP
Analyse
IMP
Analyse

genua

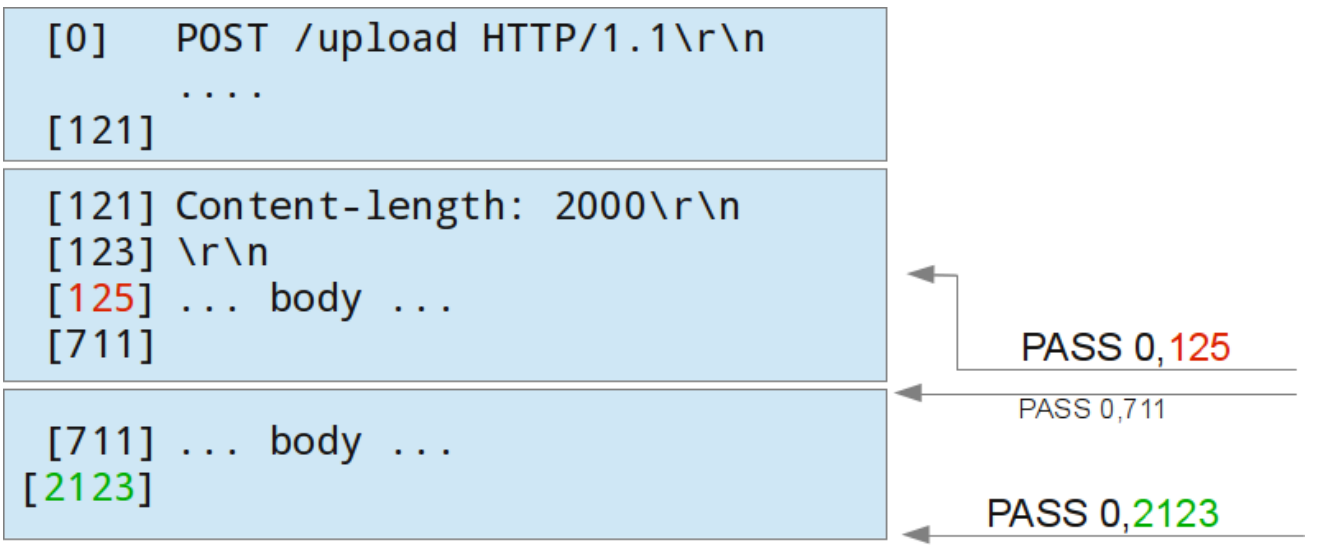
- eine Verbindung, zwei Datenströme
  - 0: Client -> Server
  - 1: Server -> Client
- Position je Datenstrom wird mitgeführt
- Anfrage an Analyse
  - Richtung, Datenchunk, [Datentyp, Position]
  - Datenquelle muss puffern, bis Antwort kommt
- Feedback per Callback
  - Durchlassen bis Position..
  - Ersetzen bis Position..
  - Blockieren der Verbindung
  - ...


0
140
280
390
700
...
PASS
PASS
REPLACE
PASS
REPLACE
0
140
200
260
350
700
...

- IMP_PASS dir,offset
- akzeptiere Daten bis Position <offset> in Richtung <dir>
- offset in Zukunft: Akzeptieren ohne Analyse
- offset == IMP_MAXOFFSET - alles weitere akzeptieren

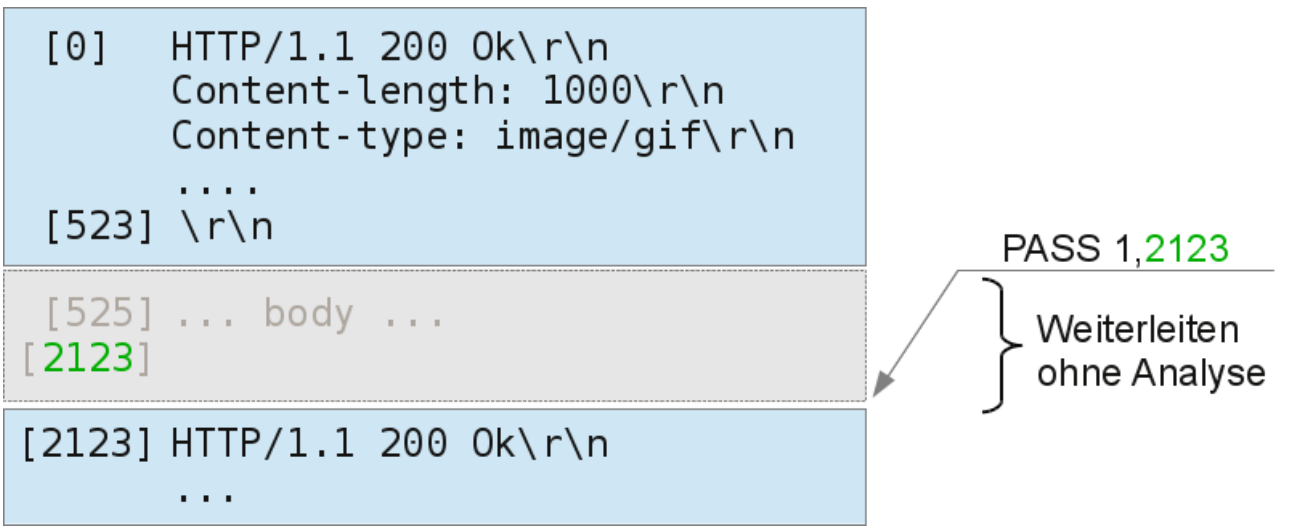
Durchlassen

Beispiel PASS dir,inside



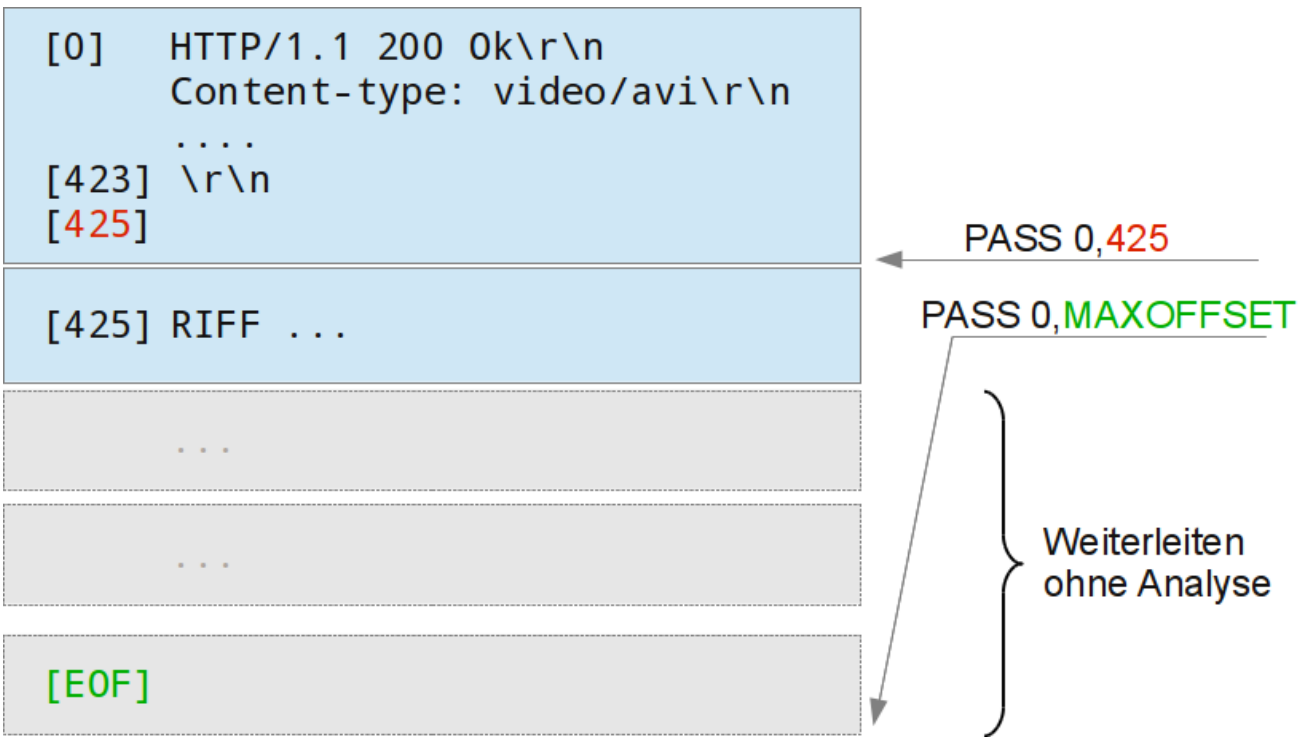
Durchlassen

Beispiel PASS dir,future



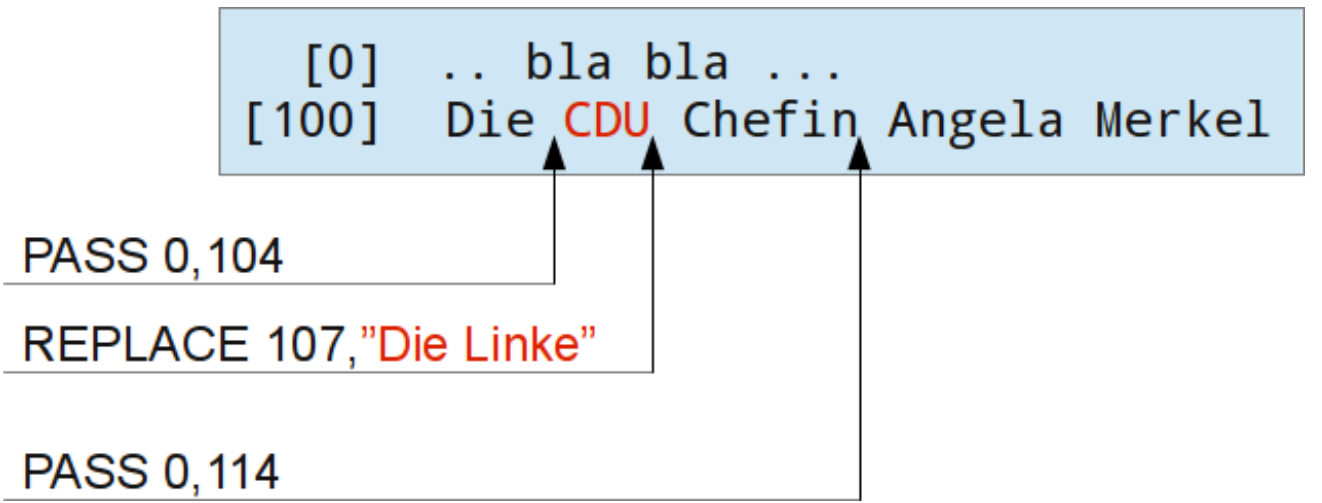
Durchlassen

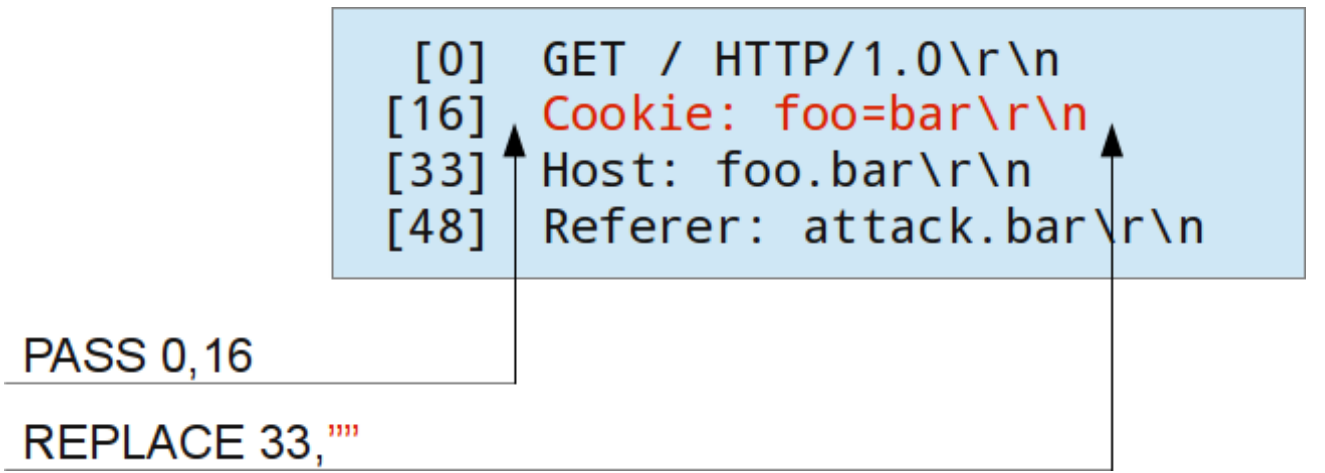
Beispiel PASS dir,infinite

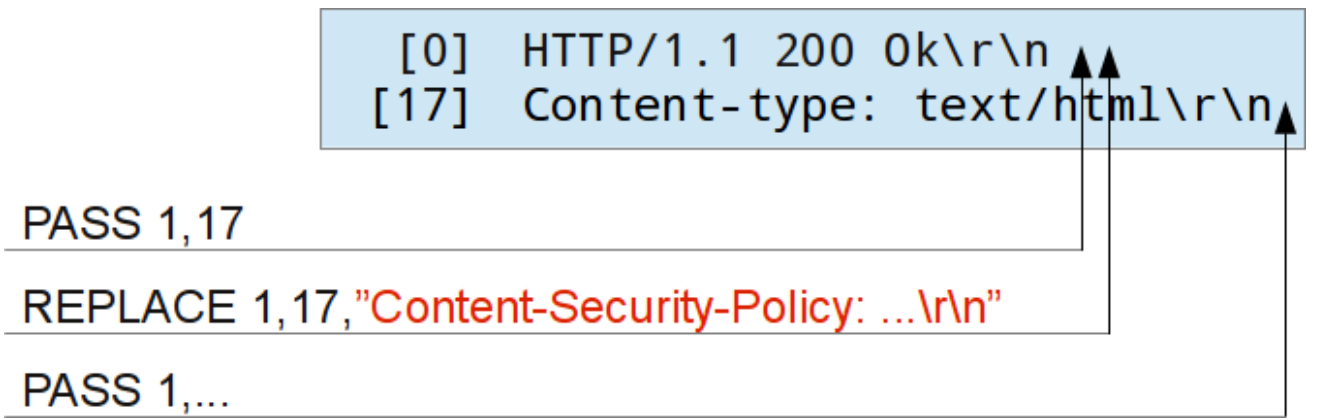


genua

- IMP_REPLACE dir,offset,newdata
- ersetze Daten bis <offset> mit <newdata>
- Startposition letzter Offset
- offset kann nicht in Zukunft sein

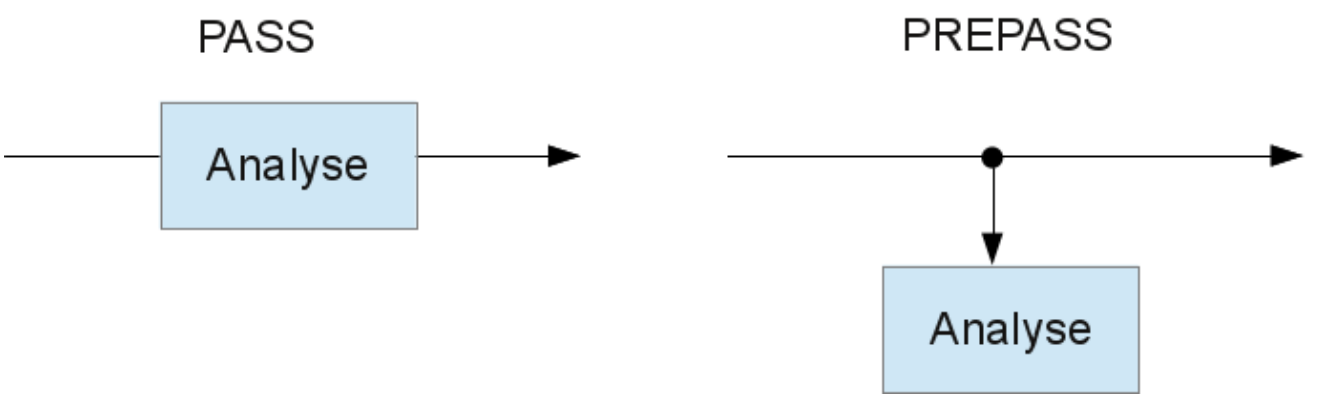

[0] .. bla bla ...
[100] Die CDU Chefin Angela Merkel
PASS 0,104
REPLACE 107,"Die Linke"
PASS 0,114


[0] GET / HTTP/1.0\r\n
[16] Cookie: foo=bar\r\n
[33] Host: foo.bar\r\n
[48] Referer: attack.bar\r\n
PASS 0,16
REPLACE 33,""


[0] HTTP/1.1 200 Ok\r\n
[17] Content-type: text/html\r\n
PASS 1,17
REPLACE 1,17,"Content-Security-Policy: ...\r\n"
PASS 1,...

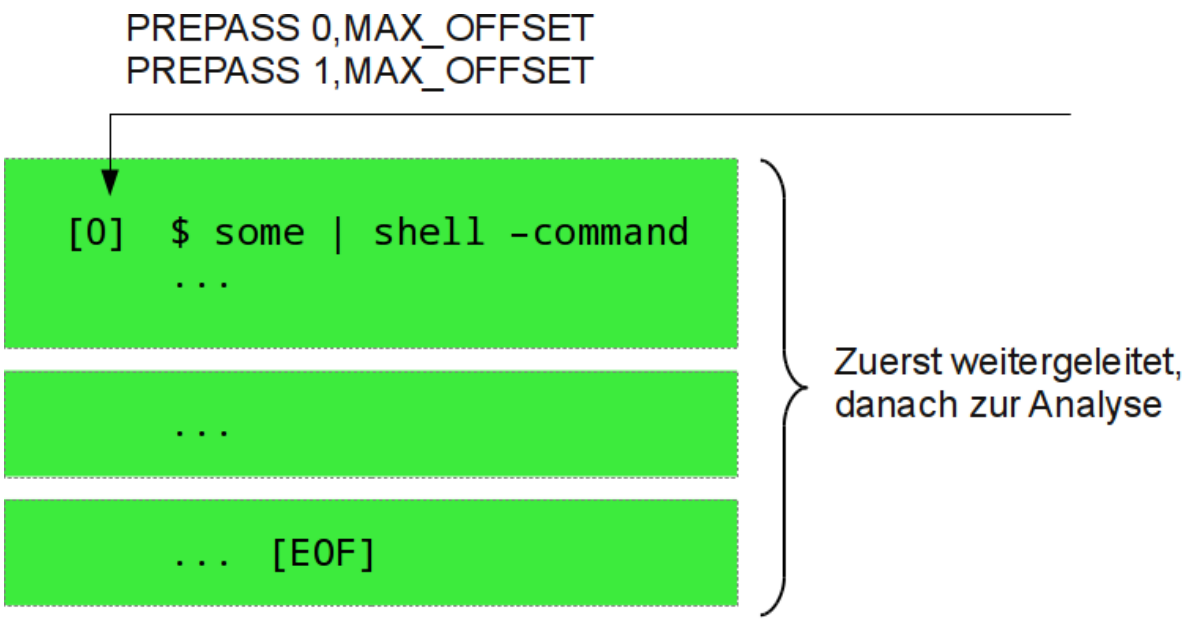
- IMP_DENY dir,reason
- Verbindung abbrechen weil <reason> in <dir>

- IMP_PREPASS dir,offset
- Durchlassen bis <offset> (Zukunft), aber trotzdem analysieren
- Anwendungen
  - Risiko vs. Latenz abwägen (window)
  - State in Analyse halten
  - nur loggen



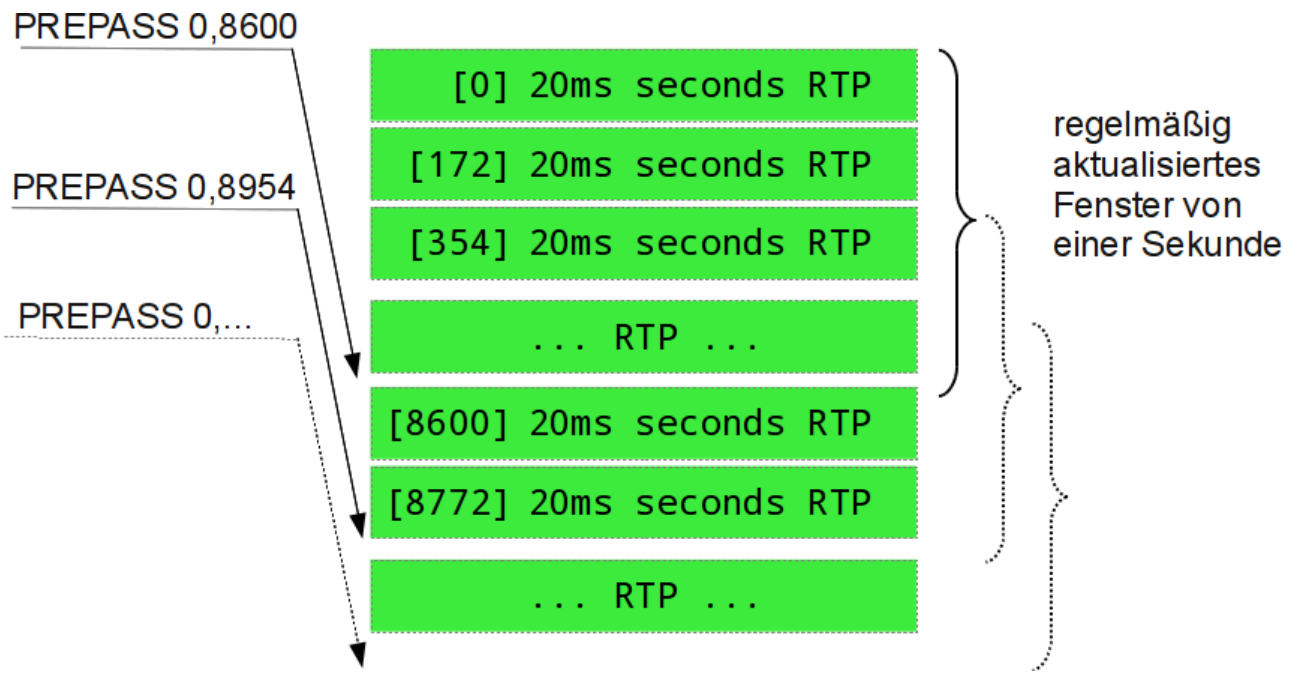
Vorab durchlassen

Beispiel PREPASS infinite - SessionLog



Vorab durchlassen

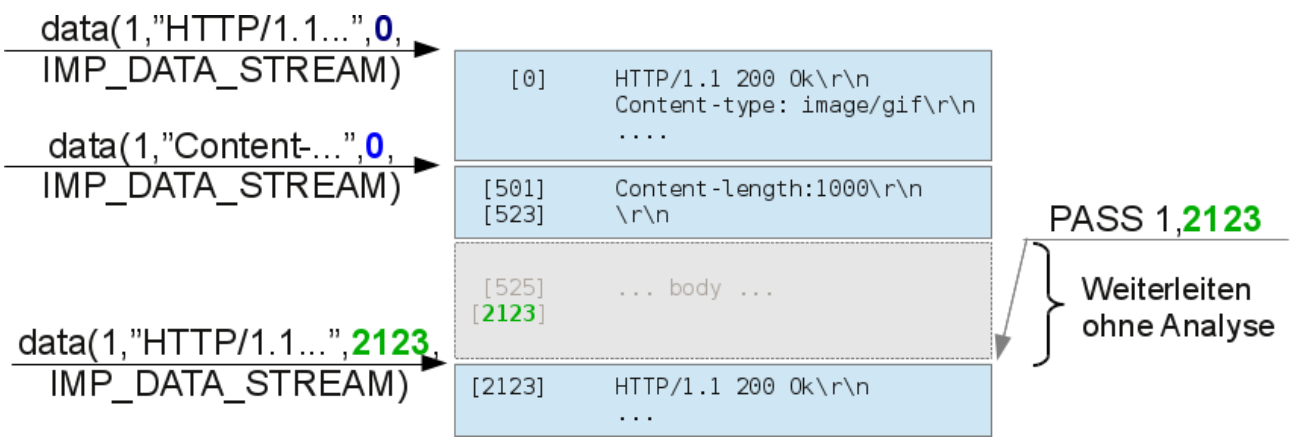
Beispiel PREPASS Window - RTP Stream



genua

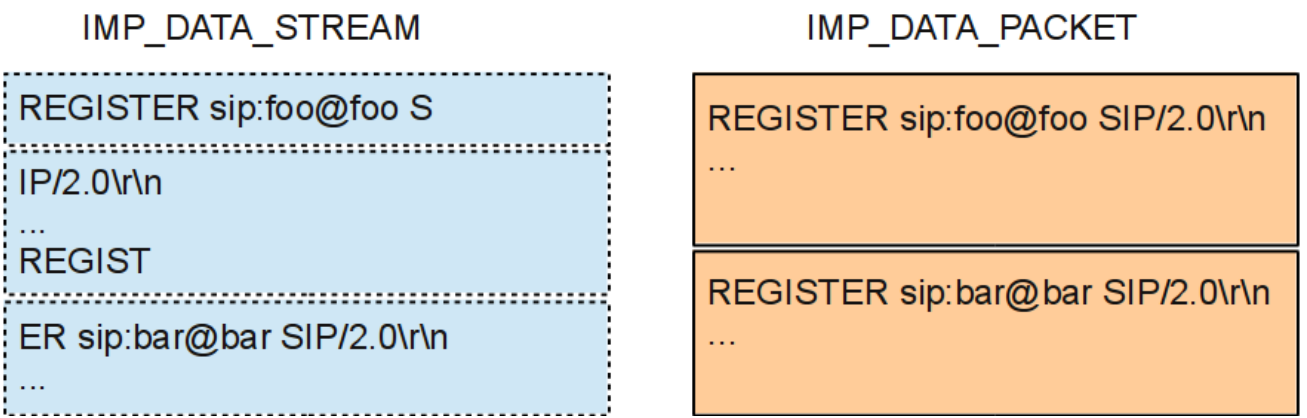
- IMP_LOG - logging
- IMP_ACCTFIELD - accounting
- IMP_TOSENDER - Daten zum Sender, z.B. bei Kommando ablehnen
- ...

- analyzer->data(dir,data,offset,type)
  - dir - die Richtung
  - data - die Daten
  - offset - ab welcher Position, nach pass <zukunft>
  - type - Datentyp



genua

- streaming Daten, Typ <= -1
  - können aneinandergehängt werden
  - teilweise Ersetzungen mittendrin möglich
  - z.B. IMP_DATA_STREAM(-1)
- Pakete, Typ >= +1
  - jedes Paket unabhängig vom anderen
  - nur im Stück durchlassen oder ersetzen
  - z.B. IMP_DATA_PACKET(+1)


IMP_DATA_STREAM
REGISTER sip:foo@foo S
IP/2.0\r\n
...
REGIST
ER sip:bar@bar SIP/2.0\r\n
...
IMP_DATA_PACKET
REGISTER sip:foo@foo SIP/2.0\r\n
...
REGISTER sip:bar@bar SIP/2.0\r\n
...

genua

- in-process Net::IMP in Perl
  - imp-relay: Proxy, Socks4, transparent(?)
  - imp-pcap-filter: pcap_in -> [IMP] -> pcap_out
  - genugate Firewall
- client-server Ansätze in OpenBSD relayd
- evtl. Nutzung in OpenFlow Controllern (SDN)
- Implementation in weiteren IDS/IPS und Programmiersprachen gern gesehen

IMP Implementierungen

existierende IMP Plugins

- Pattern - deny|replace|..
- ProtocolPinning - Protokolle forcieren
- SessionLog - Session loggen als pcap Stream
- Cascade - mehrere Plugins in Reihe
- ...

Beispiel 1: SSL Analyse

Funktionsweise

- Ziel: Server Zertifikat aus SSL-Datenstrom extrahieren
- Schritte:
  - Server Hello einlesen (Record#1)
  - Certificate Handshake finden (Record#2)
  - commonName extrahieren und loggen
- siehe Net::IMP::Example::LogServerCertificate

genua

```
package Net::IMP::Example::LogServerCertificate;
use base 'Net::IMP::Base';
use Net::IMP qw(:log :DEFAULT); # import IMP_ constants
use Net::IMP::Debug;
use fields (
     'done',  # set after Server Hello or if no SSL
     'sbuf',  # buffer data for processing Server Hello
);
```

Beispiel 1: SSL Analyse

Interface deklarieren

genua

```
sub INTERFACE {
    return ([
        IMP_DATA_STREAM,                    # what we accept
        [ IMP_PASS, IMP_PREPASS, IMP_LOG ]  # what we return
    ])
}
```

genua

```
sub new_analyzer {
    my ($factory,%args) = @_;
    my $self = $factory->SUPER::new_analyzer(%args);

    $self->run_callback(
        # we are not interested in data from client
        [ IMP_PASS, 0, IMP_MAXOFFSET ],
        # and we will not change data from server, only inspect
        [ IMP_PREPASS, 1, IMP_MAXOFFSET ],
    );

    $self->{sbuf} = '';
    return $self;
}
```

Beispiel 1: SSL Analyse

Daten füttern (1)

genua

```
sub data {
    my ($self,$dir,$data) = @_;
    return if $dir == 0; # should not happen
    return if $self->{done}; # done or no SSL
    return if $data eq ''; # eof from server

    my $buf = $self->{sbuf} .= $data;
    ....
```

Beispiel 1: SSL Analyse

Daten füttern (2)

genua

```
...
if ( _read_ssl_handshake($self,\$buf,2)                    # Server Hello
    and my $certs = _read_ssl_handshake($self,\$buf,11)  # Certificates
) {
    $self->{done} = 1;

    # find OID 2.5.4.3 (common name) the quick and dirty way
    if ( $certs =~m{\x06\x03\x55\x04\x03.}g
        and my $name = _get_asn1_string(substr($certs,pos($certs))) {
        $self->run_callback([ IMP_LOG,1,0,0,IMP_LOG_INFO,"cn=$name" ]);
    }
}

$self->run_callback([ IMP_PASS,1,IMP_MAXOFFSET ])
    if $self->{done};
```

genua

```
$ imp-pcap-filter.pl \
   -MNet::IMP::Example::LogServerCertificate \
   -r rsasnakeoil2.cap \
   -w x.pcap
[info] cn=Snake Oil CA
```

Beispiel 1: SSL Analyse
Variationen

- kein PREPASS, sondern PASS nach Prüfen Zertifikat
- nur wirkliches SSL durchlassen, einfaches Tunneling verhindern
- SSLv2 verbieten
- nur starke Verschlüsselung erlauben
- Zertifikate vom Debian Random Fauxpas ausschliessen
- Zertifikat gegen Perspectives/Convergence checken
- Zertifikatpinning

Beispiel 2 - IRC Shout

Funktionsweise

- Ziel: meine IRC Messages in GROSSBUCHSTABEN
- Weg:
  - auf Clientseite lauschen
  - PRIVMSG rcpt message abfangen
  - und mit uc(message) weiterleiten

genua

```
package IRCShout;
use Net::IMP; # import IMP_ constants
use base 'Net::IMP::Base';
use fields (
    'pos',   # current position in stream
    'line',  # buffer for unfinished lines
);
```

Beispiel 2 - IRC Shout

Interface

```
sub INTERFACE {
    return ([
        IMP_DATA_STREAM,
        [ IMP_PASS, IMP_REPLACE ]
    ])
}
```

```
sub new_analyzer {
    my ($factory,%args) = @_;
    my $self = $factory->SUPER::new_analyzer(%args);

    $self->run_callback(
        # we are not interested in data from server
        [ IMP_PASS, 1, IMP_MAXOFFSET ],
    );

    $self->{line} = '';
    $self->{pos} = 0;
    return $self;
}
```

genua

```
sub data {
    my ($self,$dir,$data) = @_;
    return if $dir == 1; # should not happen
    return if $data eq ''; # eof from client
    $self->{line} .= $data;

    my @rv; # collect callbacks
    while ( $self->{line} =~s{\A([^\n]*\n)}{} ) {
        my $line = $1;
        $self->{pos} += length($line);
        ....
```

genua

```
if ( shout(\$line)) {
   if ( @rv and $rv[-1][0] == IMP_REPLACE ) {
       # merge into last replacement
       $rv[-1][2] = $self->{pos};
       $rv[-1][3].= $line;
   } else {
       # add new replacement
       push @rv, [ IMP_REPLACE,0,$self->{pos},$line ];
   }
```

genua

```
} else {
   if ( @rv and $rv[-1][0] == IMP_PASS ) {
       # merge into last pass
       $rv[-1][2] = $self->{pos};
   } else {
       # add new pass
       push @rv, [ IMP_PASS,0,$self->{pos} ];
   }
}
```

```
    }
    $self->run_callback(@rv) if @rv;
}
```

Beispiel 2 - IRC Shout

howto shout

```
sub shout {
    my $line = shift;
    return $$line =~s{\A
        (
            (?: :\S+\x20+ )?       # opt msg prefix
            PRIVMSG\x20+\S+\x20+   # privmsg rcpt
        )
        (.+)                       # message
    }{$1\U$2}x                     # shout message
}
```