genua

HTTP Inspektion und Manipulation mit Net::IMP

Steffen Ullrich, genua mbH

Deutscher Perl-Workshop 2013, Berlin

genua

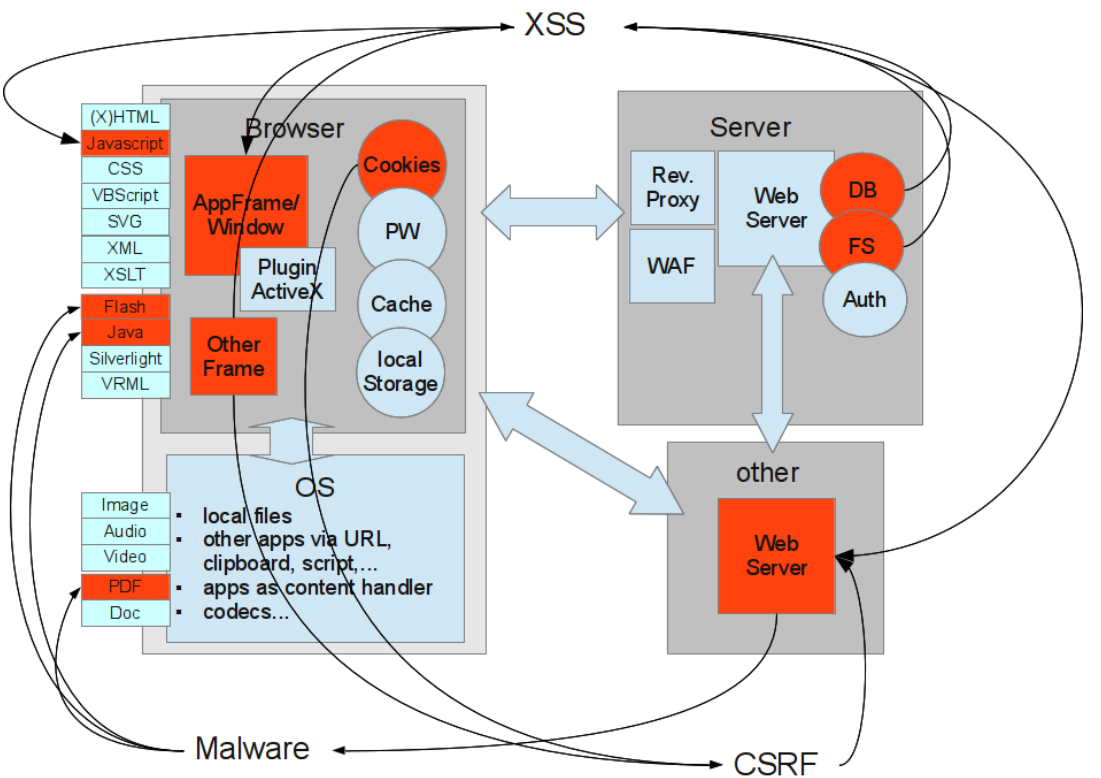
- Steffen Ullrich
- Perl-Entwickler seit 1996
  - cpan SULLR, github.com/noxxi
  - noxxi.de - Code, Talks..
- seit 2001 bei genua mbH
  - Entwicklung von Firewalls für Hochsicherheitsumgebungen
- seit 2011 Forschungsprojekt Padiofire

- was ist Padiofire
- was ist IMP
- IMP und HTTP
- existierende Implementationen
- Erstellen eigener Analysen

- Forschung zur Web 2.0 Sicherheit
- zusammen mit BTU Cottbus, FAU Erlangen, Uni Innsbruck
- gefördert vom BMBF
- Ziel: sicher bewegen in unsicherem Web
- Weg: Filtern in Perimeterfirewalls

- Malware
  - via Malvertising, Waterholing...
  - 2013 prominente Opfer: Facebook, Microsoft, Apple
  - 2013 prominente Spreader: Sparkasse, PC-Welt, NBC
- Cross-Site-Scripting (XSS)
  - Reflected|Stored|DOM XSS
  - alle mal betroffen: Yahoo, Facebook, Google, Paypal....
- Cross-Site-Request-Forgery (CSRF)
  - oft in Routern zu finden: D-Link, Netgear, Linksys...
  - 2012: 4,5 Millionen Router in Brasilien gehackt
- u.v.m

Padiofire
ausgewählte Attacken im Bild



- alle Web-Apps 100% fixen: nur ein Traum
- Web Application Firewalls (WAF): i.A. nur Anstrich, kein Schutz
- Penetration Testing: je billiger, destso weniger Probleme
- Qualitätssiegel: Snake Oil
- NGFW (Next Generation Firewalls), UTM (Unified Threat Management):
  - Marketing vs. Realität
  - etwas Schutz, aber schauen nicht tief genug
- Client-side Firewalls:
  - Heuristiken können Verhaltensauffälligkeiten finden
- im Browser: NoScript, FlashBlock, Click-To-Play, csFire, AdBlockPlus..
  - können Angriffsfläche effektiv vermindern

Padiofire
neue Ideen gebraucht

- derzeitige Lösungen reichen nicht aus
- brauchen viele neue Ideen
- wollen Rapid Prototyping von Ideen
- mit einer schnellen Überführung in diverse Produkte

genua

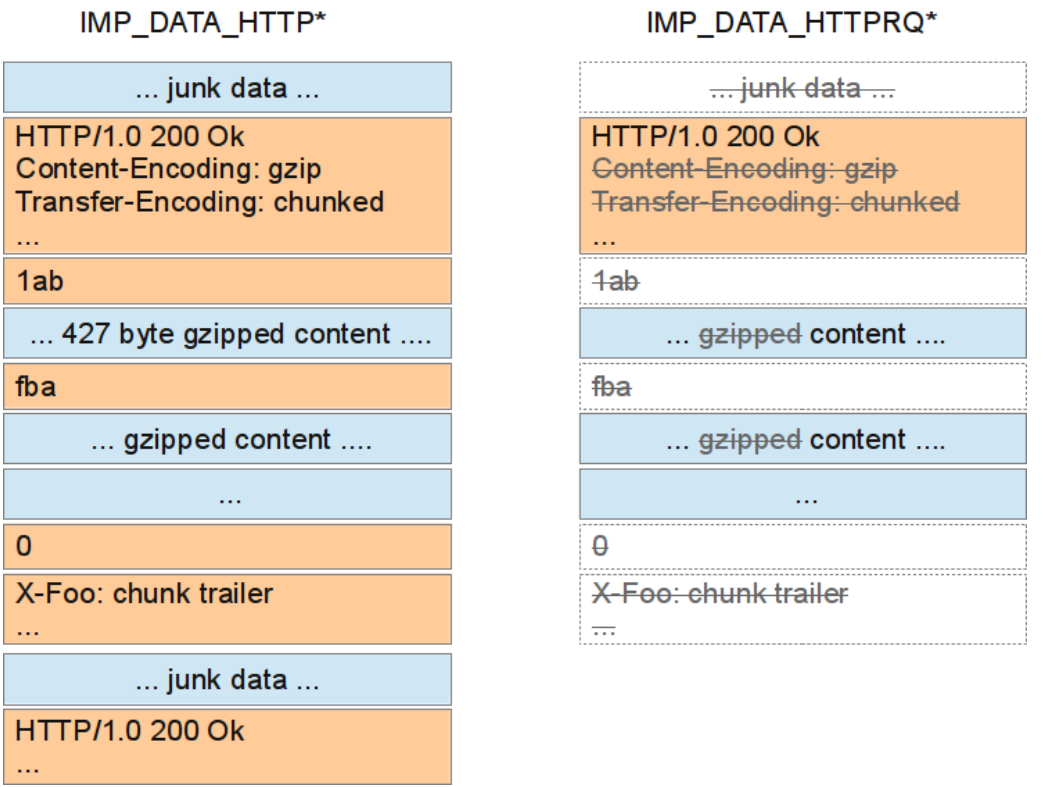
- IMP - Inspection and Modification Protocol
- Schnittstelle zwischen Analysesystemen und Analyse
  - gleiche Analyse in div. Proxies, IDS... nutzbar
- DDIM vs DPI: Deeper Data Inspection and **Modification**
- einfach, performant, streaming...
- Details siehe Vortrag zu IMP

- IMP spezifisch für HTTP
- HTTP spezifische Datentypen und vereinfachtes Interface
- Connection
  - Junk, Header, Body, Chunk-Header, Chunk-Trailer, Data
  - mehrere Requests je Connection möglich
- Request:
  - einzelner Request
  - Header
  - Content - unchunked, unkomprimiert
  - Data (Websocket, SSL)

Net::IMP::HTTP

Connection und Request im Bild

genua



genua

- App::HTTP_Proxy_IMP
  - transparente Decompression on-demand
  - http_proxy_imp --filter myAnalyzer listen-host:port
- genugate PoC

genua

- CSRFProtect: CSRF Protection
- LogFormData: loggen von Formulardaten
- SaveResponse: Sichern von Response zu URL
- FakeResponse: eigene Response für URL
- SessionLog: Speichern als pcap
- PoC Content-Security-Policy Erstellung
- Binding libpadiofire

- Net::IMP::HTTP::Example::FlipImg
- Ziel: Bilder auf den Kopf stellen
  - Request ignorieren
  - Response != gif|png|jpeg ignorieren
  - Content > 2**16 ignorieren
  - Einlesen, Flip, per IMP REPLACE weitergeben

TOUAREG XXL

VW enthüllt seinen riesigen CrossBlue

VW zeigt erstmals den neuen CrossBlue. Der stattliche, 4,99 m lange Gelände-Van überragt den großen Touareg um 19 cm.
mehr...

KURIOSES TUNING

Diese Autos (er)kennt kein Mensch

So mancher Autobastler verwandelt sich in eine Art David Copperfield des Blechs. Da mutiert schnell mal ein Toyota MR2 zum Mustang.
mehr...

EIN BÖSES EF

Tipps fü nach de

Vor dem Spiel i bringt Tücken m

LESER-REPORTER & BILD KÄMPFT

genua

```
package Net::IMP::HTTP::Example::FlipImg;
use base 'Net::IMP::HTTP::Request';
use fields (
    'ignore',  # irrelevant content
    'image',   # collected octets for image
);

use Net::IMP;  # import IMP_ constants
use Net::IMP::Debug;
use Image::Magick;
use constant MAX_SIZE => 2**16;
```

genua

```
sub RTYPES { (IMP_PASS,IMP_REPLACE) }
sub new_analyzer {
    my ($factory,%args) = @_;
    my $self = $factory->SUPER::new_analyzer(%args);
    # request data do not matter
    $self->run_callback([ IMP_PASS,0,IMP_MAXOFFSET ]);
    return $self;
}
```

```
sub request_hdr {}
sub request_body {}
sub any_data {}
```

genua

```
sub response_hdr {
    my ($self,$hdr) = @_;
    my $ignore;
    # we only want selected image/ content types and not too big
    my ($ct) = $hdr =~m{\nContent-type:[ \t]*([^\s;]+)}i;
    my ($clen) = $hdr =~m{\nContent-length:[ \t]*(\d+)}i;
    my ($code) = $hdr =~m{\AHTTP/1\.[01][ \t]+(\d+)};
    ...
```

genua

```
...
if ( $code != 200 ) {
    debug("will not rotate code=$code");
    $ignore++;
} elsif ( ! $ct or $ct !~m{^image/(png|gif|jpeg)$} ) {
    debug("will not rotate content type $ct");
    $ignore++;
} elsif ( $clen and $clen > 2**16 ) {
    debug("image is too big: $clen" );
    $ignore++;
}
...
```

```
    ...
    if ( $ignore ) {
        $self->run_callback([ IMP_PASS,1,IMP_MAXOFFSET ]);
        $self->{ignore} = 1;
        return;
    }

    # pass header
    $self->run_callback([ IMP_PASS,1,$self->offset(1) ]);
}
```

Beispiel: FlipImg

Response Body 1

```
sub response_body {
    my ($self,$data) = @_;
    $self->{ignore} and return;
    my $off = $self->offset(1);

    if ( $data ne '' ) {
        ....
```

```
...
$self->{image} .= $data;  # append to image
# replace with '' in caller to save memory there
# at the end we will replace eof with the flipped image
$self->run_callback([ IMP_REPLACE,1,$off,'' ]);

# with chunked encoding we don't get a length up front, so check now
if ( length($self->{image}) > MAX_SIZE ) {
    debug("image too big");
    $self->run_callback(
        [ IMP_REPLACE,1,$off,$self->{image} ], # unchanged image
        [ IMP_PASS,1,IMP_MAXOFFSET ]
    );
    $self->{ignore} = 1;
}

return;
...
```

```
        ...
    }

    # end of image reached, flip with Image::Magick
    debug("flip image size=%d",length($self->{image}));
    my $img = Image::Magick->new;
    debug("failed to flip img: $@") if ! eval {
        $img->BlobToImage($self->{image});
        $img->Flip;
        ($self->{image}) = $img->ImageToBlob;
        debug("replace with ".length($self->{image})." bytes");
        1;
    };

    $self->run_callback(
        [ IMP_REPLACE,1,$self->offset(1),$self->{image} ],
        [ IMP_PASS,1,IMP_MAXOFFSET ],
    );
}
```

```
http_proxy_imp --filter Example::FlipImg 127.0.0.1:8888
```

Die FDP stellt sich neu auf, um im Bundestagswahlkampf zu punkten. Gesundheitsminister Daniel Bahr sieht in Rainer Brüderle ganz klar die Nummer eins seiner Partei. Und er erklärt, was die FDP mit McDonald's und dem Dschungelcamp gemeinsam haben. »

Deutschland 415

User-Ranking

Benoten Sie Politiker!

Steuern, Afghanistan, Energiepolitik – es gibt viele strittige Themen in der Politik. Wie bewerten Sie die Akteure? Benoten Sie Kanzlerin und Kollegen in drei Kategorien. »

eutschland 27

Turbulente Suche nach neuem Bundespräsidenten

Merkel musste Trittin nach Gaucks Nummer fragen

Es war ein ziemliches Geschacher um das höchste Amt im Staat: Nachdem Joachim Gauck am Ende doch als neuer Bundespräsidentenkandidat feststand, wollte Angela Merkel ihn anrufen – doch die Kanzlerin

www.focus.de/politik/deutschland/politiker-ranking/user-ranking-benoten-sie-politiker_aid_360690.html

Zur Startseite

Zinsvergleich O
biallo.de/Baufinanz
Baufinanzierungs-
unverbindlich!

Alles zur Energ

Deutschland 28.02.2

Biblis nach Fuku

Akw-Stilllegu Hessen 200

Die Stilllegung des A das hessische Umw Fukushima-Katastro des Verwaltungsger Jetzt droht dem Lan Schadenersatzklage

Weniger Extrawürste
Ökostrom-Umlage-Sch zahlen

Beispiel: Deny per Content-Type

Ziel

- filtern basierend auf Content-type der Response
- Aber u.a. bei script und evtl. style includes wird content-type ignoriert

Beispiel: Deny per Content-Type

Includes etc

genua

```
package Net::IMP::HTTP::Example::BlockContentType;
use base 'Net::IMP::HTTP::Request';
use Net::IMP;  # import IMP_ constants
use Net::IMP::Debug;
```

genua

```
sub validate_cfg {
    my ($class,%cfg) = @_;
    my @err;
    for my $k (qw(whiterx blackrx)) {
        my $rx = delete $cfg{$k} or next;
        ref($rx) and next;
        push @err,"$k is no valid regexp: $@" if ! eval { qr/$rx/ };
    }
    return (@err,$class->SUPER::validate_cfg(%cfg));
}
```

Beispiel: Deny per Content-Type

Config aus String

```
sub str2cfg {
    my ($class,$str) = @_;
    my %cfg = $class->SUPER::str2cfg($str);
    for my $k (qw(whiterx blackrx)) {
        next if ! $cfg{$k} or ref $cfg{$k};
        $cfg{$k} = eval { qr/$cfg{$k}/ }
            or die "invalid rx in $k: $@";
    }
    return %cfg;
}

http_proxy_imp \
  --filter 'Example::BlockContentType=whiterx=^text/&blackrx=...' \
  127.0.0.1:1235
```

Beispiel: Deny per Content-Type

neuer Analyzer

genua

```
sub RTYPES { ( IMP_PASS, IMP_DENY ) }
sub new_analyzer {
    my ($factory,%args) = @_;
    my $self = $factory->SUPER::new_analyzer(%args);
    # request data do not matter
    $self->run_callback([ IMP_PASS,0,IMP_MAXOFFSET ]);
    if ( ! $self->{factory_args}{whiterx}
        && ! $self->{factory_args}{blackrx} ) {
        # nothing to analyze
        $self->run_callback([ IMP_PASS,1,IMP_MAXOFFSET ]);
    }
    return $self;
}
```

Beispiel: Deny per Content-Type

Dummies

```
sub request_hdr {}
sub request_body {}
sub response_body {}
sub any_data {}
```

Beispiel: Deny per Content-Type

Response Header 1/4 - get content-type

genua

```
sub response_hdr {
    my ($self,$hdr) = @_;
    # we only want selected image/ content types and not too big
    my $ct = $hdr =~m{\nContent-type:[ \t]*([^\s;]+)}i && lc($1)
        || 'unknown/unknown';

    my $reason;
    ...
```

Beispiel: Deny per Content-Type

Response Header 2/4 - check white list

```
...
if ( my $white = $self->{factory_args}{whiterx} ) {
    if ( $ct =~ $white ) {
        debug("allowed $ct because of white list");
        goto pass;
    } else {
        debug("denied $ct because not in white list");
        $reason = "denied $ct because not in white list";
        goto deny;
    }
}
...
```

Beispiel: Deny per Content-Type

Response Header 3/4 - check black list

genua

```
...
if ( my $black = $self->{factory_args}{blackrx} ) {
    if ( $ct =~ $black ) {
        debug("denied $ct because in black list");
        $reason = "denied $ct because in black list";
        goto deny;
    } else {
        debug("allow $ct because not in black list");
        goto pass;
    }
}
...
```

Beispiel: Deny per Content-Type

Response Header 4/4 - do pass/deny

```
    ...
    pass:
    $self->run_callback([ IMP_PASS,1,IMP_MAXOFFSET ]);
    return;

    deny:
    $self->run_callback([ IMP_DENY,1,$reason ]);
    return;
}
```

Beispiel: Deny per Content-Type

mögliche Verbesserungen

- auf Accept-Header des Requests schauen
  - text/css - will css
  - image/* - will image
  - aber: bei Redirect vergisst FF den Kontext
- per File::MMagic o.ä. raten
- "X-Content-Type-Options: nosniff" gegen IE injizieren

- Content-Filter mit File::MMagic
- URL-Filter, z.B. mit AdBlockPlus Listen
- Virenscanner per ICAP, auch für File-Uploads
- POST XML Filter (SOAP)
- div Analysen HTML, Script..
- Script/IFrame Injection (Ads, XSS-Protection)
- jegliche Amazon Zugriffe mit Affilate Tags versehen
- ...