Zur Funktionsweise von Schadsoftware

Stefan Schumacher

Magdeburger Institut für Sicherheitsforschung

Chemnitzer Linux-Tage 2013



Über mich

- Direktor des Magdeburger Instituts für Sicherheitsforschung
- IT-Sicherheitsberater mit Fokus auf Social Engineering, Security Awareness, Counter Intelligence
- Hacker seit knapp 20 Jahren, u.a. NetBSD-Entwickler



Inhaltsverzeichnis

1. Einführung
2. Geschichte
3. Malware allgemein
4. Angriffsvektoren
5. Was muss Malware können?
6. Gegenmaßnahmen



Book



Kap. 3: Timeo Danaos et dona ferentes - Zur Funktionsweise von Schadsoftware

Hypothese

Ich nutze Apple/Linux/OpenBSD/FreeBSD/NetBSD/Solaris, ich bin sicher vor Viren!



Hypothese

Ich nutze Apple/Linux/OpenBSD/FreeBSD/NetBSD/Solaris, ich bin sicher vor Viren!

Nö, nicht wirklich. Ich zeige euch warum.



Hypothese

Ich nutze Apple/Linux/OpenBSD/FreeBSD/NetBSD/Solaris, ich bin sicher vor Viren!
Nö, nicht wirklich. Ich zeige euch warum.



Inhaltsverzeichnis

1 Einführung

2 Geschichte

3 Malware allgemein

4 Angriffsvektoren

5 Was muss Malware können?

6 Gegenmaßnahmen



1966 John von Neumann beschreibt die theoretischen Grundlagen selbstreproduzierender Programme. Sein Buch basiert auf Vorlesungen die er bereits in den 1940er Jahren gehalten hat.

1971 Bob Thomas schreibt *Creeper*, welcher DEC PDP-10-Rechner mit TENEX infiziert.

1981 Der 14-jährige Richard Skrenta veröffentlicht *Elk Cloner*. Der Virus verbreitet sich auf Apple][-Systemen über die Startdisketten. Elk Cloner gilt als erster großer Malware-Ausbruch in der freien Wildbahn.

1984 Ken Thompson erhält den Turing-Award. In seiner Rede »Reflections on Trusting Trust« beschreibt er wie man über Manipulationen am Compiler eine Backdoor in Programmen einbaut.

1986 Der Pakistani-Virus wird von Basit Farooq Alvi und Amjad Farooq Alvi veröffentlicht. Er gilt als der erste IBM-PC-kompatible Virus und ist verantwortlich für die erste Epidemie auf PCs.



1987 Mit SCA verbreitet sich der erste größere Amiga-Virus über den Bootsektor.

1988 Robert Tappan Morris entwirft den Morris-Wurm und entlässt ihn am MIT in die Wildbahn. Der Wurm nutzte verschiedene bekannte Sicherheitslücken in Unix-Diensten aus und infizierte DEC VAX-Systeme mit 4BSD oder Sun-3. Man geht davon aus, dass der Wurm ca. 6 000 Rechner infizierte, was 10% aller Rechner im damaligen Internet entsprach. Durch die entstehende Last auf den infizierten Systemen und im Netzwerk wurde das damalige Internet nahezu lahmgelegt.

1989 Fridrik Skulason entdeckt Ghostball, den ersten multipartiten Virus, d.h. Virus der sich auf verschiedenen Wegen verbreitet.

1994 Mit OneHalf wird der erste polymorphe Virus veröffentlicht. Er verändert bei jeder Infektion seine eigene Signatur, um die Erkennungsrate zu senken.



1995 Der erste Makro-Virus wird veröffentlicht, er greift Microsoft Word-Dateien an.

2001 Sadmind attackiert Microsoft IIS und Sun Solaris-Systeme.

2001 Ramen-Wurm attackiert Red Hat 6.2 und 7 über rpc.statd und wuftpd.

2004 Santy, der erste Webwurm wird entdeckt. Er infiziert Webseiten die phpBB nutzen, eine Software die ein webbasiertes Diskussionsforum bereitstellt. Der Wurm nutzte Google um Server mit phpBB zu finden, bis Google die Suchanfrage blockierte.

2006 OSX/Leap-A, die erste Malware für Mac OS X wird entdeckt.

2012 Flashback greift OS X über Java an.



begriffen?

- Vulnerability: Sicherheitslücke in Software
- Exploit: Programm das eine Vulnerability ausnutzt
- Virus: vom Anwender nicht kontrollierte Änderungen am System, benötigt Wirts-Datei
- Wurm: dringt aktiv in Systeme ein z.B. über Exploits in Servern
- Trojaner: Schadroutine/-programm getarnt in nützlichem Programm
- Rootkit: Programme um die Anwesenheit von Schadsoftware, Backdoors etc. zu verbergen



Was ist Sicherheit?

Informatik: VIVA-Kriterien

Vertraulichkeit Daten bleiben geheim, Zugriff nur für Berechtigte

Integrität Daten werden nicht verändert, bleiben im Originalzustand

Verfügbarkeit Daten/Systeme sind verfügbar wenn benötigt

Authentizität Daten stammen vom vorgeblichen Absender

auf elektronischen/Computer-Systemen nicht sicherzustellen

erfordert zwingend Kryptographie!

gute Kryptographie :-)



Was ist Sicherheit?

Informatik: VIVA-Kriterien

Vertraulichkeit Daten bleiben geheim, Zugriff nur für Berechtigte

Integrität Daten werden nicht verändert, bleiben im Originalzustand

Verfügbarkeit Daten/Systeme sind verfügbar wenn benötigt

Authentizität Daten stammen vom vorgeblichen Absender

auf elektronischen/Computer-Systemen nicht sicherzustellen
erfordert zwingend Kryptographie!
gute Kryptographie :-)



Inhaltsverzeichnis

1 Einführung

2 Geschichte

3 Malware allgemein

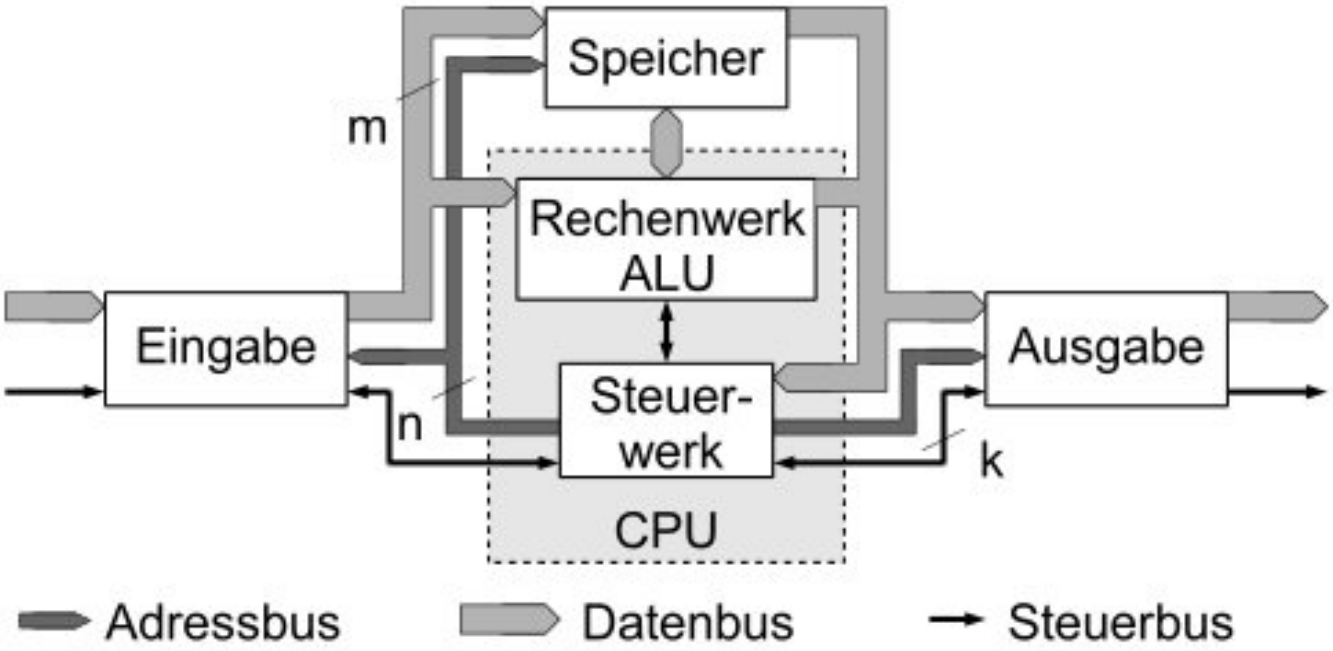
4 Angriffsvektoren

5 Was muss Malware können?

6 Gegenmaßnahmen



von-Neumann-Architektur



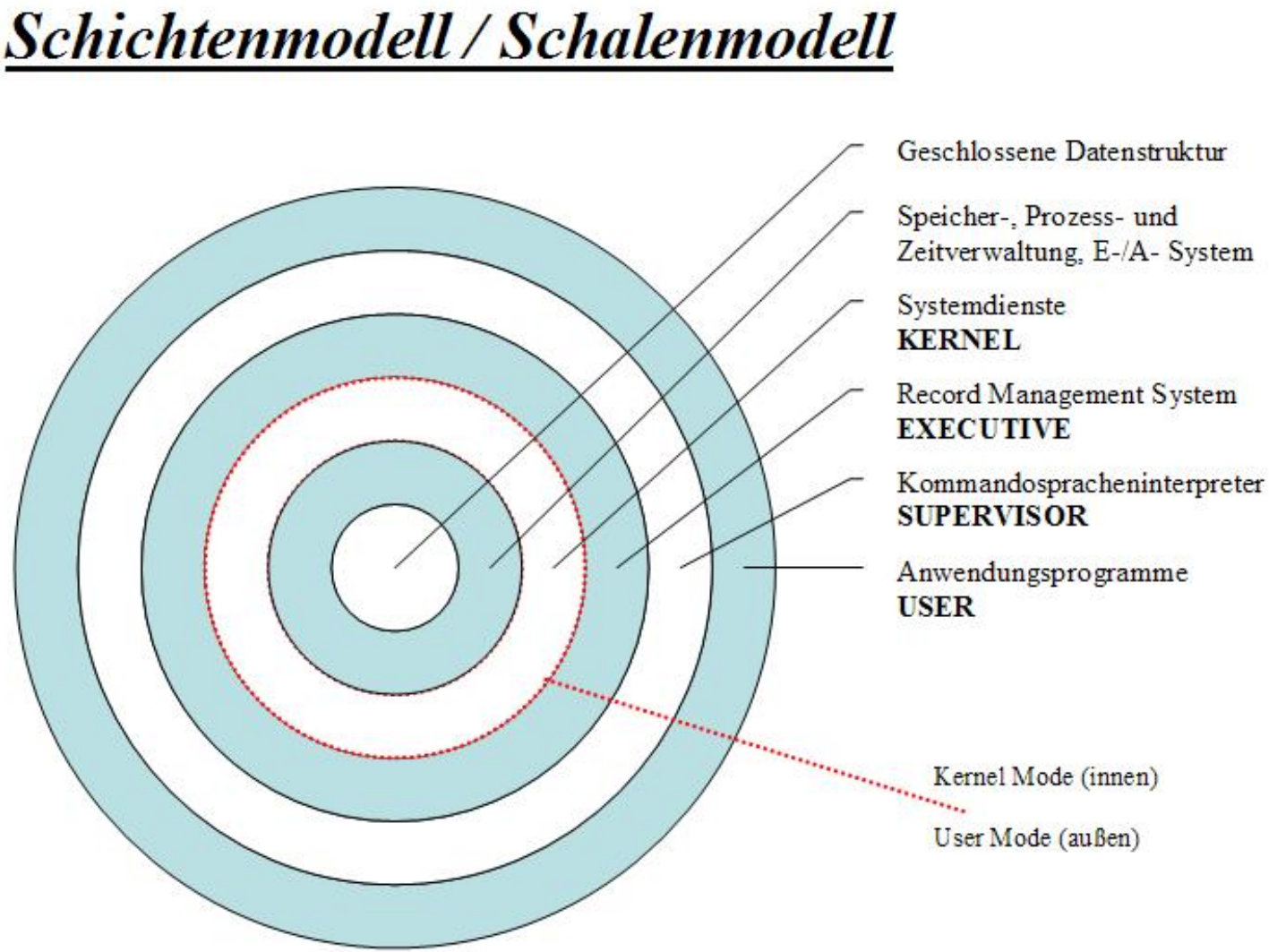
- grundlegende Computer-Architektur, Turing-Maschine
- ein Speicher enthält Befehle und Daten
- deterministischer Programmablauf garantiert, Race-Conditions und Daten-Inkohärenzen ausgeschlossen
- ABER: ein Programm kann u.U. auf fremde Daten zugreifen

Benutzerrechte

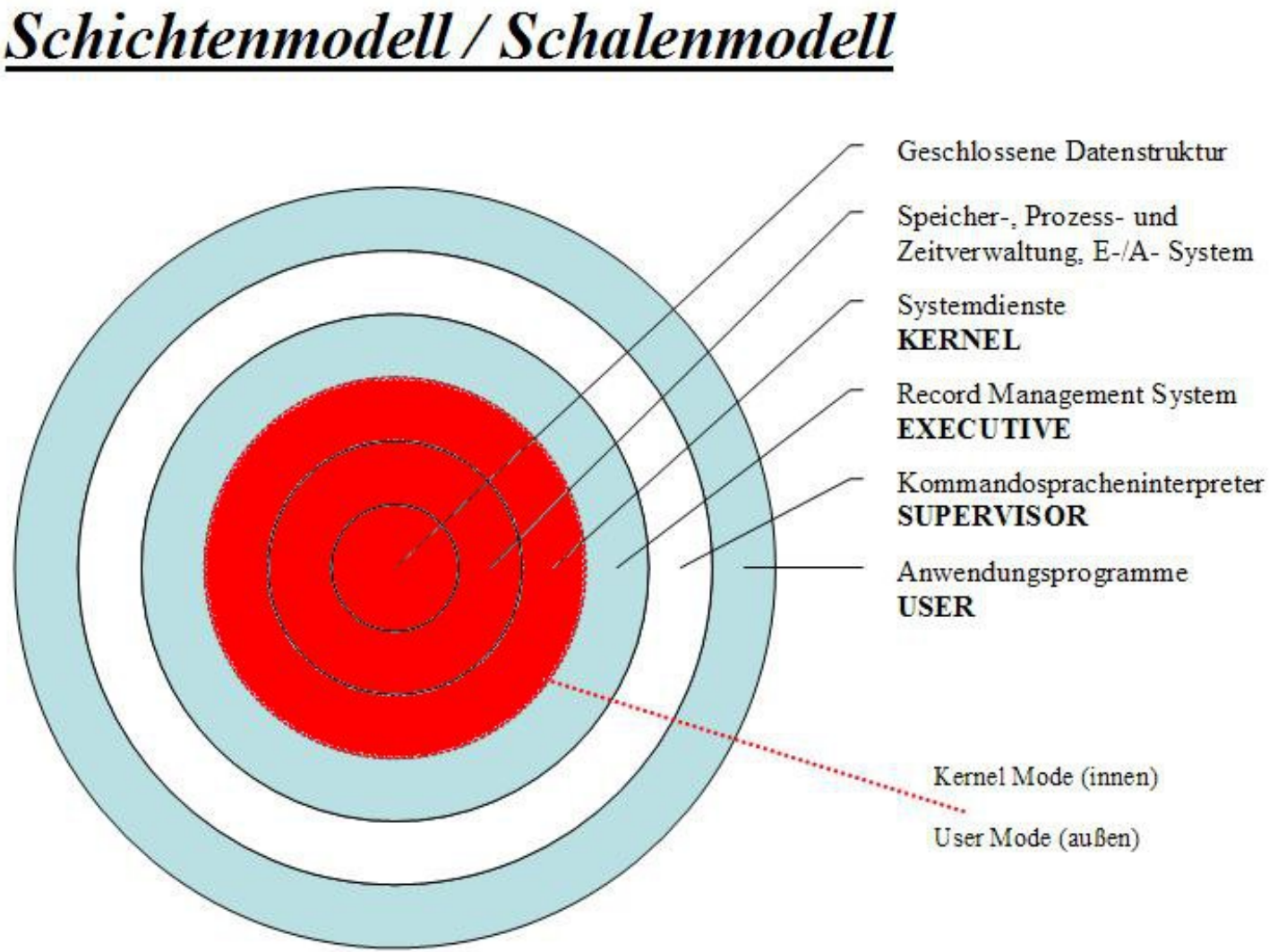
- `$ shutdown -h now`
  `shutdown: NOT super-user`
- `$ apachectl stop`
  `This operation requires root.`
- `$ rm /etc/hosts`
  `override rw-r--r-- root/wheel for /etc/hosts? y`
  `rm: /etc/hosts: Permission denied`



Schalenmodell



Schalenmodell



setuid

- Einige Programme benötigen Root-Rechte
- su, sudo, cron, at, chsh, Apache etc.
- gesamtes Programm läuft mit Root-Rechten $\rightsquigarrow$ Gefahr
- Apache benötigt nur einige Systemcalls mit Root-Rechten (binden an Port 80)
- übernehme ich Apache, kann ich /etc/master.passwd auslesen ...
- Idee: Systemcalls feiner granulieren mit Systrace
- Nur die notwendigen Systemcalls laufen mit Root-Rechten
- s. Magdeburger Journal zur Sicherheitsforschung



Inhaltsverzeichnis

1 Einführung

2 Geschichte

3 Malware allgemein

4 Angriffsvektoren

5 Was muss Malware können?

6 Gegenmaßnahmen



eher historisch

- Master Boot Record – Disketten :-)
- VBS/Skript/Makro-Viren für MS Office
- Outlook Express
- Würmer über unsicher Dienste: SQL Slammer, Nimda, RTM ...



Trojaner ausrollen

- Trojaner: Schadsoftware die als Nutzlast einer anderen Software mitreist
- Tarnsoftware wird vom Anwender installiert, Schadsoftware im Hintergrund ebenfalls (Spear Phishing :-)
- Installation durch physikalischen Zugriff (Flughafenkontrolle, Wohnungseinbruch)
- gezielter entfernter Einbruch durch Ausnutzen von Sicherheitslücken
- Privilege Escalation



Login

- Remote-Login über telnet, ftp, SSH etc.
- Passwörter raten/kapern
- Ziel: Root-rechte bekommen



Beliebte Stratfor-Passwörter

Passwort	Häufigkeit
stratfor	12023
123456	625
0000	519
password	517
stratfor1	426
strat4	265
changeme	265
1qaz2wsx	232
1234	228
wright	179
usmcportal	148
abc123	89
qwerty	85
12345	75
12345678	74



Gegenmaßnahme

- starke Passwörter verwenden
- Remote Logins einschränken, kein ftp/telnet/rsh
- SSH: IP-Adressen filtern, Root-Login verbieten, Schlüssel-Login, Log-Dateien lesen, ”komische” Namen verwenden
- siehe Youtube.de/Sicherheitsforschung



Inhaltsverzeichnis

1 Einführung

2 Geschichte

3 Malware allgemein

4 Angriffsvektoren

5 Was muss Malware können?

6 Gegenmaßnahmen



Nach der Installation

- Schadsoftware muss sich verstecken vor
  Betriebssystem, Virenscanner, Anti-Spyware, Intrusion Detection
- mit Root-Rechten geht das
- mit Root-Rechten geht alles :-)
- Rootkits können jeden Datenstrom mitlesen und archivieren

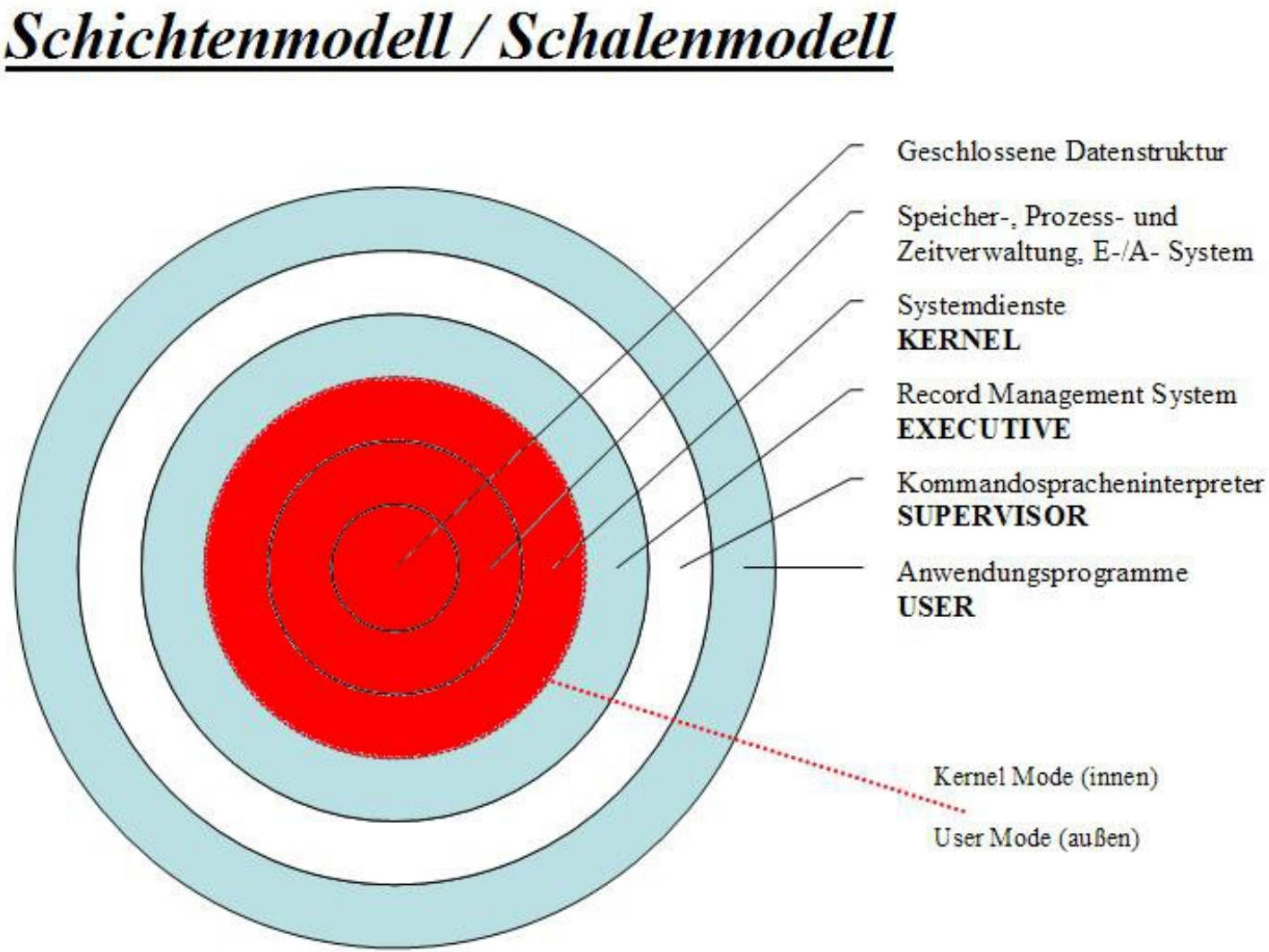


Nach der Installation

- Schadsoftware muss sich verstecken vor
  Betriebssystem, Virenscanner, Anti-Spyware, Intrusion Detection
- mit Root-Rechten geht das
- mit Root-Rechten geht alles :-)
- Rootkits können jeden Datenstrom mitlesen und archivieren



Schalenmodell



Was muss eine gute Malware können?

- unerkannt bleiben (Stealth: Prozesse, Nutzer, Dateien verstecken auf NetBSD/Alpha)
- Updates einspielen können (Nachladefunktion)
- Daten sammeln und abschicken, möglichst unerkannt (Kryptographie/Stenographie)
- Daten vor ihrer Verschlüsselungen/Schlüssel abschnorcheln
- Dateien nach Schlüsselwörtern durchsuchen (RegEx)
- Screenshots machen
- fernsteuerbar durch C&C-Server (Remote-Login)



Inhaltsverzeichnis

1. Einführung
2. Geschichte
3. Malware allgemein
4. Angriffsvektoren
5. Was muss Malware können?
6. Gegenmaßnahmen



Hostbasierte IDS

- AIDE, Tripwire, Mtree erstellen Fingerabdruck des Systems mit krypt. Prüfsummen
- ABER: das Opfersystem ist nicht vertrauenswürdig -¿ Live-CD o.ä. booten
- ex falso sequitur quodlibet!
- Siehe CLT 2005

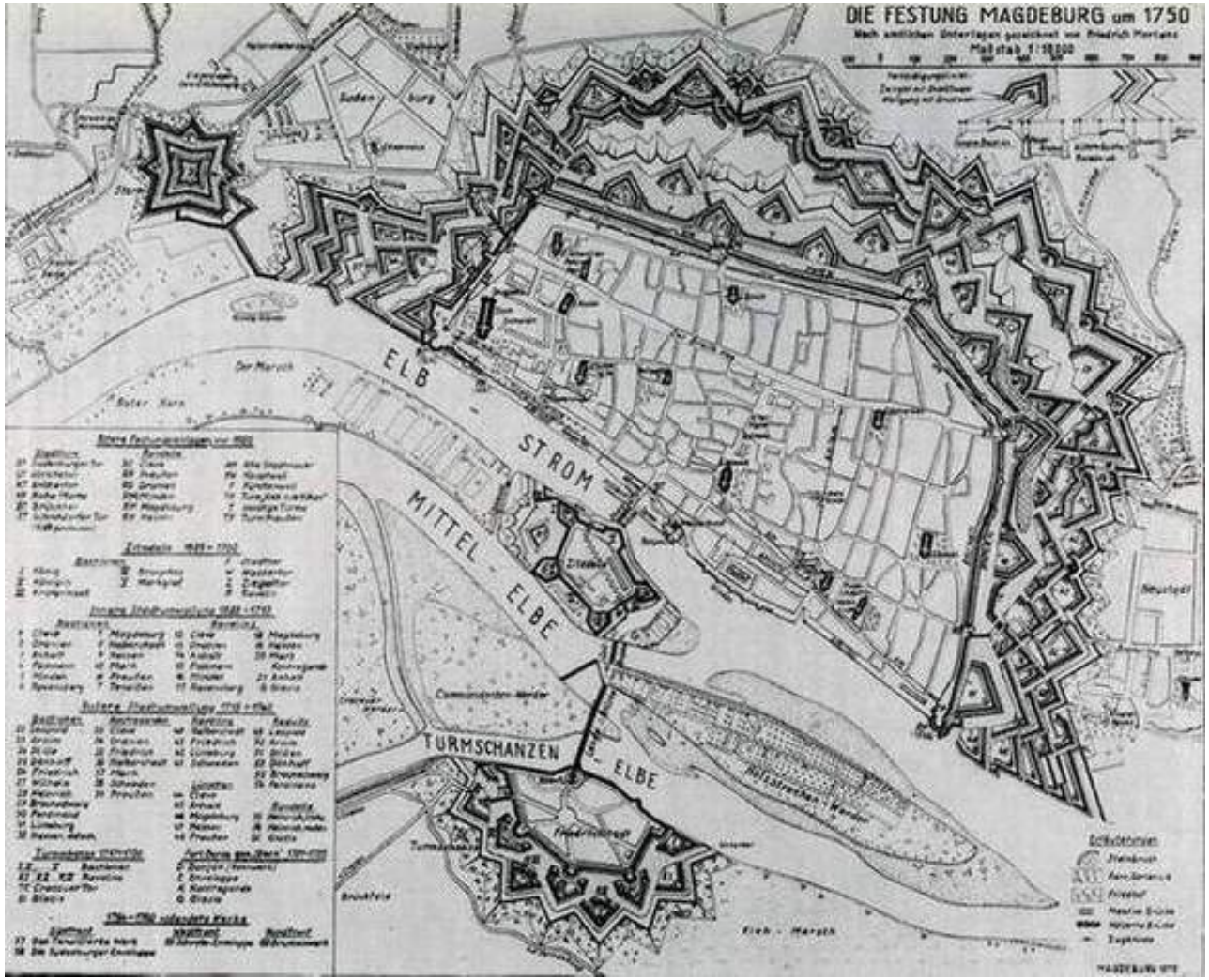


Virenscaner

- clamav: freier Virenscanner
- ab und zu mal drüberlaufen lassen



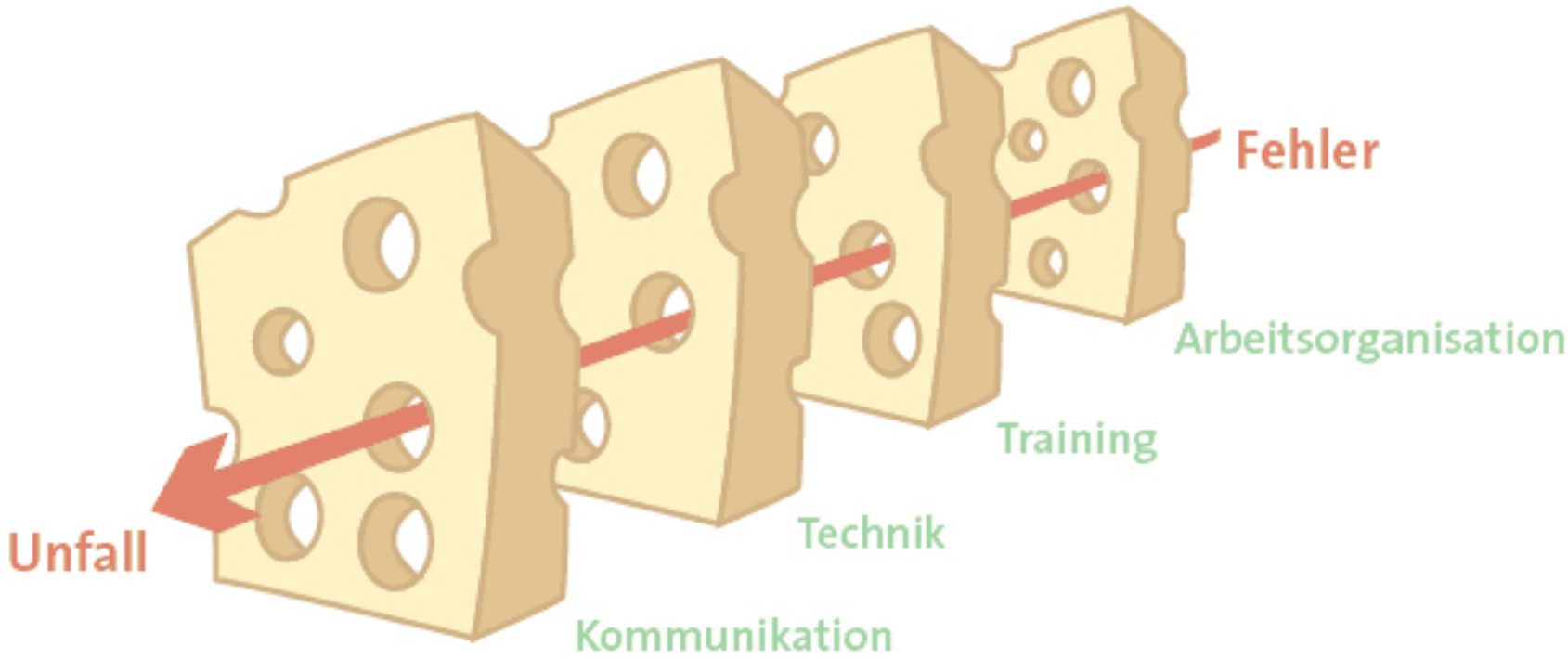
Schalenmodell



Die Feste Magdeburg um 1750: Verteidigung in der Tiefe

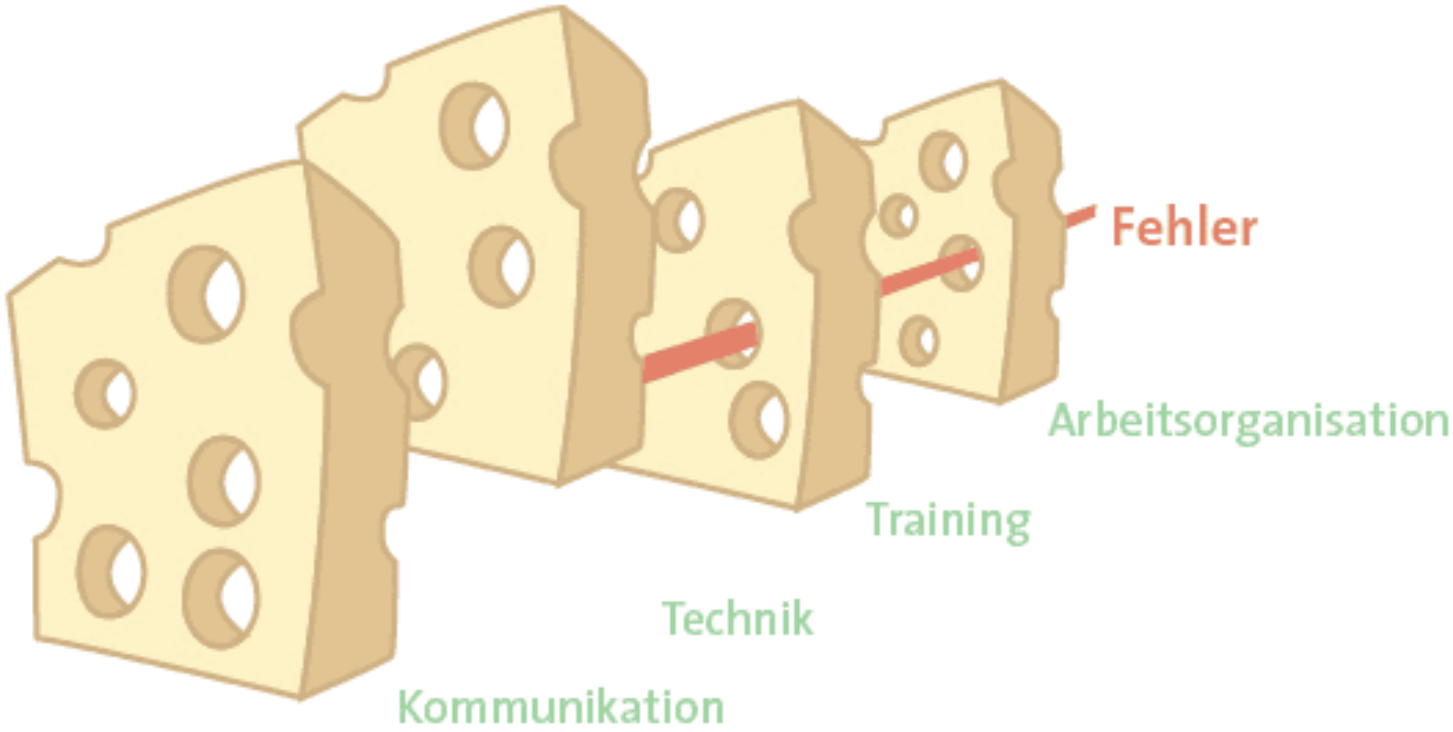
Die Verteidigung in der Tiefe angewandt auf IT-Systeme: Secrétariat général de la défense nationale, Direction centrale de la sécurité des

Schalenmodell



Reason, James: ”The Contribution of Latent Human Failures to the Breakdown of Complex Systems”

Schalenmodell



Reason, James: ”The Contribution of Latent Human Failures to the Breakdown of Complex Systems”

Sonstige Trojaner

- Carrier IQ
- Steam/Valve
- Sony BMG DRM
- Facebook, Google Desktopsuche ...
- Payback ...
- einfach mal die EULA lesen ;-)



Sonstige Trojaner

- Carrier IQ
- Steam/Valve
- Sony BMG DRM
- Facebook, Google Desktopsuche ...
- Payback ...
- einfach mal die EULA lesen ;-)



Joanna Rutkowska

Upcoming Security Nightmares ...

- Rootkits und CPU-Virtualisierung
- mit Alexander Tereshkin und Rafal Wojtczuk: 0wning XEN
  Rootkit im XEN-Hypervisor via DMA $\rightsquigarrow$ Ring 0
  kaum nachweisbar!
- Blue Pill Rootkit: versteckt sich dank Virtualisierung, ist kaum aufspürbar



Gegenmaßnahmen

- Rechner nicht aus der Hand geben
- Laptop ohne wichtige Daten mitnehmen
- BIOS-Passwort
- Verschlüsselung der Festplatte (IBM TPM)
- Verschlüsselung des Dateisystems (Truecrypt, CGD, GBDE, EncFS, CryptoFS)
- Fingerabdruck des Systems erstellen und vergleichen (mtree, aide, tripwire)
- Image des Systems ziehen und zurückspielen :-) (G4U, Ghost for Unix, feyrer.de/g4u)



Gegenmaßnahmen

- Systrace einsetzen um System Calls zu regeln
- Qubes OS http://qubes-os.org
- Open Source rules!
- Never trust an Operating System you don't have sources for :-)



Gegenmaßnahmen

- Systrace einsetzen um System Calls zu regeln
- Qubes OS http://qubes-os.org
- Open Source rules!
- Never trust an Operating System you don't have sources for :-)



Literatur

- `www.Cryptomancer.de`
  Projektseite zur Kryptographie
- Stefan Schumacher: *Einführung in kryptographische Methoden* (online)
- Stefan Schumacher: *Verschlüsselte Dateisysteme für NetBSD* GUUG UpTimes 4/2006
- Stefan Schumacher: *Kryptographische Dateisysteme im Detail* in: Tagungsband Chemnitzer Linux-Tage 2011

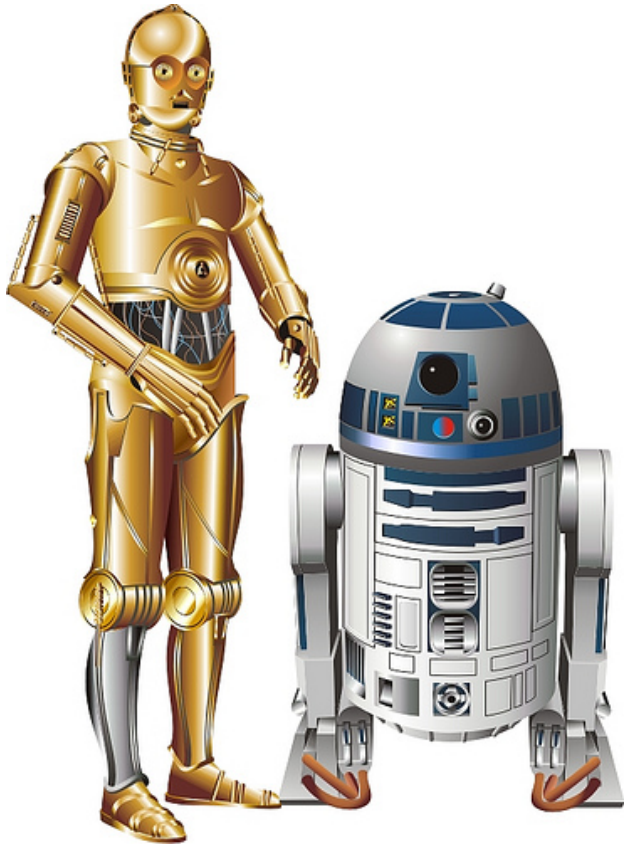


Literatur

- Stefan Schumacher: *Einbruchserkennung in Netzwerke mit Intrusion Detection Systemen und Honeypots* (online)
- Stefan Schumacher: *Auf dem Weg zum Intrusion Detection System der nächsten Generation* in: Tagungsband Chemnitzer Linux-Tage 2010
- Stefan Schumacher: *Systeme mit Systrace härten* Datenschleuder 91
- Stefan Schumacher: *Psychologische Grundlagen des Social-Engineering* Datenschleuder 94
- Feyrer H., S. Schumacher & M. Weinem: *NetBSD – Das portabelste Betriebssystem der Welt*. In: Open Source Jahrbuch (2007)
- Stefan Schumacher *Sicherheit messen. Eine Operationalisierun als latentes soziales Konstrukt* In: Die sicherheitspolitische Streitkultur in der Bundesrepublik Deutschland (2011) (aka Goldene Eule)



die Besten der Besten der Besten, Sir!



The city's central computer told you?
R2D2, you know better than to trust a strange computer!

Fragen?

Mail: stefan.schumacher@sicherheitsforschung-magdeburg.de
9475 1687 4218 026F 6ACF 89EE 8B63 6058 D015 B8EF

Jabber: stefan.schumacher@guug.de
youtube.de/Sicherheitsforschung

