

## Contents

1. Editorial
2. scip AG Informationen
3. Neue Sicherheitslücken
4. Statistiken Verletzbarkeiten
5. Labs
6. Bilderrätsel
7. Impressum

### 1. Editorial

#### Des Teufels Zahl

Ich mag Zahlen. Und ich mag es, mit ihnen zu spielen: Zahlen sind eine wohldefinierte Klasse von Symbolen, die je nach angewendetem metrischen und arithmetischen System gewissen nachvollziehbaren Eigenarten unterworfen sind. In der Regel repräsentieren Zahlen eine Sache oder einen Zustand. Die dezimale Zahl 5 steht für die Anzahl meiner Finger an einer Hand. Und die dezimale Zahl 186 definiert meine Körpergrösse in Zentimeter. Zahlen abstrahieren also die Realität. Sie können sie weder ersetzen noch auf den Kopf stellen. Sie können sie lediglich, und dies oftmals nur mit erheblichem Aufwand, abbilden. In vielen Fällen gar unter zwingender Zuhilfenahme von Reduktion (z.B. Rundung) nur skizzieren.

Zahlenmystik (Numerologie) erachte ich als Unsinn: Solange die Zu- und Abneigung bezüglich gewissen Zahlen regional und kulturell unterschiedlich aufgefasst werden, solange kann ich Zahlen keine Emotionen entgegenbringen: In der Babylonische Zahlensymbolik ist die 14 die Zahl der bösen Dämonen, in China meidet man die 9 - die auch für Beerdigung steht - und in westlichen Ländern fürchtet man die 13. Die letztgenannte Triskaidekaphobie wurde übrigens erst spät, nämlich erstmals im Jahr 1869 beschrieben. Die emotionale Bindung zu Zahlen ist damit genauso unsinnig, wie ein Streitgespräch über Glauben. Denn ohne Wissen

kann keine Wahrheit gefunden werden und ohne die Möglichkeit des Findens von Wahrheit birgt ein vermeintlicher Diskurs keinen Nutzen in sich.

Ich stelle nun die waghalsig erscheinende Frage, ob gewisse Zahlen verboten werden sollen. In den USA finden sich viele Hochhäuser ohne einen dreizehnten Stock. Er ist damit ein Tabu und könnte ja eigentlich auch gesetzlich verboten werden. Den Buchhalter wirds jedoch nicht freuen, da er bei einem positiven Saldo zwischen 12 und 14 entscheiden müsste (das Rundungsverhalten in diesem Fall müsste man also auch gesetzlich regeln; reelle Zahlen können in diesem Fall eine Approximation anstreben). Aber ein Schutz vor bösen Geistern auf Kosten von Genauigkeit? Das ist nicht der Sinn eines deterministischen Zahlensystems.

Ein generelles Verbot einer Zahl macht also keinen Sinn. Man sollte, falls überhaupt, die Einschränkung lediglich auf spezifische Bereiche festlegen. Kein dreizehnter Stock, dann haben wir wenigstens schon mal etwas kleines erreicht. Doch was ist nun, wenn wir eine 1'405 Zahl verbieten würden? Es scheint (bisher) unsinnig, diese beim Bau von Wolkenkratzern zu berücksichtigen. Doch wir können sie im Computerbereich verbieten lassen. So geschehen bei der von Phil Carmody entwickelten Implementierung der DeCSS-Entschlüsselung, welche genutzt wurde, um den Kopierschutz von DVDs zu umgehen.

In der digitalen Welt nehmen Zahlen unweigerlich eine wichtige Rolle ein. Im Binärsystem werden Daten durch eine Dualität, bestehend aus 0 und 1, wiedergegeben. Diese Informationen können nach Belieben in andere Zahlensysteme umgewandelt werden. Traditioneller Weise werden oktale und hexadezimale Systeme verwendet. Oder man nutzt das gängige Dezimalsystem, um die gleichen Daten darzustellen.

Wir müssten also auch die Varianten aller anderen Zahlensysteme verbieten lassen. Doch was ist nun, wenn genau diese Zahl - im Übrigen eine Primzahl - in anderem Zusammenhang genutzt wird? Zum Beispiel zur Berechnung der Statik eines Hauses? Oder wenn es sich hierbei um die Darstellung eines elektronischen Fotos mit einer neuartigen Komprimierung handelt? Ist das Foto dann auch verboten? Jenes Foto,

welches (zufälligerweise?) die gleiche Datengrundlage wie der verbotene Code hat?

Juristen werden nun argumentieren, dass das Motiv im Zweifelsfall darüber entscheidet, ob es sich um eine Straftat handelt oder nicht. Wenn ich also ein Bildformat entwickle, das zur Darstellung des Empire State Building (Foto siehe Wikipedia) genau die gleiche Datenrepräsentation wie der DeCSS-Code benötigt, dann bewege ich mich im legalen Rahmen. Dass man halt dann mit der Eingabe von `"mv empire.pix decss.c; gcc -o decss decss.c; chmod +x decss; ./decss"` einen ursprünglich verbotenen Code "generieren" und ausführen kann, das war dann halt Pech/Glück (je nach Standpunkt).

Der Gedanke muss nun weitergesponnen werden, wie es sich denn zum Beispiel mit Kinderpornografie verhält, wenn diese in einem Format gespeichert wird, zu dem es (offiziell) keinen Viewer/Converter gibt. Macht sich jemand strafbar, wenn er Datenmüll hortet, den nur er in fragwürdiges Material zurückverwandeln kann.

Und was ist, wenn der Täter effektiv kein Programm hierfür benötigt und dementsprechend auch keines besitzt, da er durch Verständnis des Datenformats die Ursprungsdaten in seinem Kopf "rekonstruieren" kann? Halt genauso, wie jemand, der der Source Code einer Software durchschaut und zu jedem Punkt genau weiss, welchen Zustand das Benutzerinterface und die jeweiligen Datenbereiche haben werden. Durch klassische Konditionierung nach Pawlow, die wohl über Jahre durchgesetzt werden müsste, liessen sich simple Symbole mit Emotionen verknüpfen. Der Buchstabe G könnte sodann für eine verbotene Handlung stehen. Die Ansicht dessen oder gar nur der Gedanke an ihn könnte die gewünschten Gefühle auslösen (z.B. Freude und Wollust).

In diesem Fall ginge es nicht mehr darum Daten zu verbieten, sondern Gedanken an diese bzw. an verbotene Dinge. Dies führt zur alten Debatte darüber, ob schon alleine der Gedanke an eine Kindsmisshandlung wie eine solche geahndet (sei es auch nur durch eine medizinische / psychologische Behandlung) werden soll. Und wie sieht es aus mit einem Mord? Einem Diebstahl? Einer Lüge? Wo hört Unrecht auf und wo fängt Recht an? Und kann man diese beiden Dinge von Zahlen abhängig machen?

Marc Ruef <maru-at-scip.ch>  
Security Consultant  
Zürich, 15. März 2010

## 2. scip AG Informationen

### 2.1 Backdoor Test

Das Ziel unserer Dienstleistung Backdoor Test ist die erfolgreiche Kompromittierung der Zielumgebung durch die Infektion eines eigens angefertigten Trojanischen Pferds (Backdoor) zur Bestimmung effektiv ausnutzbarer Schlupflöcher im bestehenden Sicherheitsdispositiv.

- Vorbereitung: Die Zielumgebung wird ausgewertet, um ein individuelles Angriffsszenario entwickelt.
- Entwicklung: Ein Trojanisches Pferd wird für den Kunden programmiert. Wir bauen dabei auf unsere eigenen Code Libraries und Exploiting Payloads. SAP, iPhone, Web 2.0/Ajax, Windows Mobile, Word, Excel, PowerPoint, PDF, Outlook, Lotus Notes etc.
- Infektion: Die Zielumgebung oder ein definiertes Zielsystem wird mit dem Trojanischen Pferd infiziert (z.B. Social Engineering, Drive-By Infection, Exploiting einer Dokumentenschwachstelle).
- Fernsteuerung: Nach erfolgreicher Infektion wird die Fernsteuerung durchgesetzt, um die Machbarkeit und Möglichkeiten zu demonstrieren.

Solcherlei Backdoor Inside/Out Tests sind sehr individuell. Die Vorbereitungen (Entwicklung der Hintertür) sowie die Durchführung des Angriffs (Infektion und Fernsteuerung) werden detailliert dokumentiert. Die ausgenutzten Schwächen der Zielumgebung (z.B. Firewall-Tunneling, Antivirus Evasion, etc.) werden ausführlich besprochen.

Dank unserer langjährigen Erfahrung und unserem ausgewiesenen Expertenwissen haben wir als scip AG bereits eine Vielzahl von Backdoor Test Projekte durchgeführt.

Zählen auch Sie auf uns!

<http://www.scip.ch/?firma.referenzenalle>

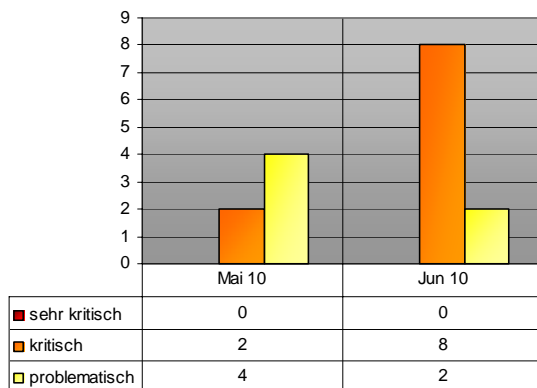
Zögern Sie nicht und kontaktieren Sie unseren Herrn Chris Widmer unter der Telefonnummer +41 44 404 13 13 oder senden Sie im eine Mail an [chris.widmer@scip.ch](mailto:chris.widmer@scip.ch).

### 3. Neue Sicherheitslücken

Die erweiterte Auflistung hier besprochener Schwachstellen sowie weitere Sicherheitslücken sind unentgeltlich in unserer Datenbank unter <http://www.scip.ch/?vuldb> einsehbar.



Die Dienstleistungspakete scip( pallas liefern Ihnen jene Informationen, die genau für Ihre Systeme relevant sind.



#### Contents:

- 4141 Apple Mac OS X verschiedene Schwachstellen
- 4140 Sophos Anti-Virus "NtQueryAttributesFile()" Privilege Escalation Schwachstelle
- 4139 Microsoft Windows Help and Support Center URL Processing Schwachstelle
- 4138 Google Chrome verschiedene Schwachstellen
- 4137 Microsoft Internet Explorer verschiedene Schwachstellen
- 4136 Microsoft Windows Media Decompression Schwachstellen
- 4135 Microsoft Windows Kernel-Mode Drivers verschiedene Schwachstellen
- 4133 Microsoft Office COM Object Instantiation Validation Schwachstelle
- 4132 Apple Safari verschiedene Schwachstellen

#### 3.1 Apple Mac OS X verschiedene Schwachstellen

Risiko: **kritisch**

Remote: Ja

Datum: 16.06.2010

scip DB: <http://www.scip.ch/?vuldb.4141>

Mac OS X (offizielle Sprechweise: Mac OS Zehn, vom altrömischen X für 10) ist ein vom Unternehmen Apple entwickeltes Betriebssystem. OS X ist die aktuelle Version aus der Produktlinie der Mac OS-Betriebssysteme für die hauseigenen Macintosh-Computer. Es ist eine proprietäre Distribution des frei erhältlichen Darwin-Betriebssystems von Apple. OS X basiert als zweites Apple-Betriebssystem (nach A/UX) auf Unix und stellt damit dessen bisher erfolgreichste kommerzielle Variante auf dem Markt für Personal Computer dar. Es kommt in abgewandelter Form beim Smartphone iPhone, dem iPad und dem tragbaren Medienabspielgerät iPod touch zum Einsatz. Die Firma Apple beschreibt in einem Advisory diverse Schwachstellen in aktuellen Versionen des Betriebssystems. Durch die Ausnutzung der Schwachstellen kann ein Angreifer möglicherweise Kontrolle über ein System erlangen.

#### Expertenmeinung:

Die diversen Schwachstellen, die von Apple in diesem kumulativen Update geschlossen werden, sind teils als kritisch zu betrachten weshalb das Einspielen des Updates als essentiell zu betrachten ist.

#### 3.2 Sophos Anti-Virus "NtQueryAttributesFile()" Privilege Escalation Schwachstelle

Risiko: **problematisch**

Remote: Ja

Datum: 10.06.2010

scip DB: <http://www.scip.ch/?vuldb.4140>

Sophos ist ein internationales Unternehmen, das Sicherheitssoftware entwickelt und diese vertreibt. Dazu gehören Virenschutz, Datenschutz, Verschlüsselungssoftware, Schutz vor Spam, Phishing, Adware, Spyware und Malware für den Unternehmensbereich sowie Universitäten und andere öffentliche Einrichtungen. Der Researcher Cody Pierce der Firma TippingPoint DV Labs beschreibt in einem Advisory eine Schwachstelle (Pufferüberlauf) in aktuellen Versionen der Applikation. Diese Schwäche erlaubt es einem Angreifer, Controller über das System zu erlangen und seine Rechte

zu erweitern.

#### Expertenmeinung:

Die vorliegende Schwachstelle ist als kritisch zu betrachten. Falls nicht bereits gesehen, sollten betroffene Benutzer und Institutionen ein zeitnahes Update über die entsprechend implementierten Mechanismen anstreben.

### 3.3 Microsoft Windows Help and Support Center URL Processing Schwachstelle

Risiko: **kritisch**

Remote: Ja

Datum: 10.06.2010

scip DB: <http://www.scip.ch/?vuldb.4139>

Microsoft Windows ist ein Markenname für Betriebssysteme des Unternehmens Microsoft. Ursprünglich war Microsoft Windows eine grafische Erweiterung des Betriebssystems MS-DOS (wie beispielsweise auch GEM oder PC/GEOS). Inzwischen wurde dieser Entwicklungszweig zugunsten der Windows-NT-Produktlinie aufgegeben und Windows bezeichnet das Betriebssystem als Ganzes. Der Name Windows (engl.: Fenster) rührt daher, dass aktive Anwendungen als rechteckige Fenster auf dem Bildschirm dargestellt werden. Der Researcher Tavis Ormandy beschreibt in einem Advisory eine Schwachstelle (Cross Site Scripting (XSS)) in aktuellen Versionen der Applikation. Durch die Ausnutzung der vorliegenden Schwachstelle kann unter Umständen beliebiger Code zur Ausführung gebracht werden, was zur Kompromittierung des Systems führen kann.

#### Expertenmeinung:

Die vorliegende Schwachstelle führte zu diversen Kontroversen, nachdem diverse Printmedien Ormandys Arbeitgeber in die laufende Diskussion einbrachten und so die Diskussion um die Schwachstelle faktisch ad absurdum führten. Defakto handelt es sich jedoch um eine kritische Schwachstelle, die zeitnahe mittels der verfügbaren Workarounds sowie des Einspielens entsprechender Patches mitigiert werden sollten.

### 3.4 Google Chrome verschiedene Schwachstellen

Risiko: **kritisch**

Remote: Ja

Datum: 09.06.2010

scip DB: <http://www.scip.ch/?vuldb.4138>

Google Chrome ist ein Webbrowser, der von Google Inc. entwickelt wird und seit dem 2.

September 2008 verfügbar ist. Am 11. Dezember 2008 erschien die erste finale Version. Zentrales Konzept ist die Aufteilung des Browsers in optisch und prozesstechnisch getrennte Browser-Tabs. Die Firma Google veröffentlichte unlängst ein Advisory, indem er eine Schwachstelle (Pufferüberlauf) in verschiedenen Versionen des Produktes beschreibt. Ein Angreifer kann durch die vorliegende Schwachstelle beliebigen Code zur Ausführung bringen und somit die Kompromittierung des Systems anstreben.

#### Expertenmeinung:

Die diversen Schwachstellen in Chrome, die Google im vorliegenden Sammeladvisory beschreibt, sollten ernst genommen und als kritisch betrachtet werden. Anwender, die Chrome einsetzen, sollten zeitnah ein Update auf eine aktuelle Version anstreben.

### 3.5 Microsoft Internet Explorer verschiedene Schwachstellen

Risiko: **kritisch**

Remote: Ja

Datum: 08.06.2010

scip DB: <http://www.scip.ch/?vuldb.4137>

Der Internet Explorer (offiziell Windows Internet Explorer; früher Microsoft Internet Explorer; Abkürzung: IE oder MSIE) ist ein Webbrowser vom Softwarehersteller Microsoft für dessen Betriebssystem Windows. Seit Windows 95B ist der Internet Explorer fester Bestandteil dieser Betriebssysteme. Bei älteren Windows-Versionen kann er nachinstalliert werden. Die aktuelle Version ist Internet Explorer 8. Der Researcher Peter Vreugdenhil veröffentlichte unlängst ein Advisory, indem er eine Schwachstelle (Pufferüberlauf) in verschiedenen Versionen des Produktes beschreibt. Durch die Ausnutzung der vorliegenden Schwachstelle kann unter Umständen beliebiger Code zur Ausführung gebracht werden, was zur Kompromittierung des Systems führen kann.

#### Expertenmeinung:

Die vorliegenden Schwachstellen sind als kritisch zu betrachten und sollten zeitnah durch das Einspielen entsprechender Patches adressiert werden.

### 3.6 Microsoft Windows Media Decompression Schwachstellen

Risiko: **kritisch**

Remote: Ja

Datum: 08.06.2010

scip DB: <http://www.scip.ch/?vuldb.4136>

Microsoft Windows ist ein Markenname für Betriebssysteme des Unternehmens Microsoft. Ursprünglich war Microsoft Windows eine grafische Erweiterung des Betriebssystems MS-DOS (wie beispielsweise auch GEM oder PC/GEOS). Inzwischen wurde dieser Entwicklungszweig zugunsten der Windows-NT-Produktlinie aufgegeben und Windows bezeichnet das Betriebssystem als Ganzes. Der Name Windows (engl.: Fenster) rührt daher, dass aktive Anwendungen als rechteckige Fenster auf dem Bildschirm dargestellt werden. Der Researcher Yamata Li der Firma Palo Alto Networks veröffentlichte unlängst ein Advisory, indem er eine Schwachstelle (Pufferüberlauf) in verschiedenen Versionen des Produktes beschreibt. Durch die Ausnutzung der vorliegenden Schwachstelle kann unter Umständen beliebiger Code zur Ausführung gebracht werden, was zur Kompromittierung des Systems führen kann.

#### Expertenmeinung:

Die vorliegenden Schwachstellen sind als kritisch zu betrachten und sollten zeitnah durch das Einspielen entsprechender Patches mitigiert werden.

### 3.7 Microsoft Windows Kernel-Mode Drivers verschiedene Schwachstellen

Risiko: **kritisch**  
 Remote: Ja  
 Datum: 08.06.2010  
 scip DB: <http://www.scip.ch/?vuldb.4135>

Microsoft Windows ist ein Markenname für Betriebssysteme des Unternehmens Microsoft. Ursprünglich war Microsoft Windows eine grafische Erweiterung des Betriebssystems MS-DOS (wie beispielsweise auch GEM oder PC/GEOS). Inzwischen wurde dieser Entwicklungszweig zugunsten der Windows-NT-Produktlinie aufgegeben und Windows bezeichnet das Betriebssystem als Ganzes. Der Name Windows (engl.: Fenster) rührt daher, dass aktive Anwendungen als rechteckige Fenster auf dem Bildschirm dargestellt werden. Der Researcher Sebastien Renaud identifizierte unlängst eine Schwachstelle (Pufferüberlauf) in aktuellen Versionen der vorliegenden Applikation. Durch die Ausnutzung der vorliegenden Schwachstelle kann unter Umständen beliebiger Code zur Ausführung gebracht werden, was zur Kompromittierung des Systems führen kann.

#### Expertenmeinung:

Die vorliegenden Schwachstellen sind als kritisch zu betrachten, da sie eine Remote Code Execution theoretisch ermöglichen. Auch hier sei an dieser Stelle empfohlen, die entsprechenden Patches des Herstellers zeitnah einzuspielen.

### 3.8 Microsoft Office COM Object Instantiation Validation Schwachstelle

Risiko: **kritisch**  
 Remote: Ja  
 Datum: 08.06.2010  
 scip DB: <http://www.scip.ch/?vuldb.4133>

Microsoft Office ist das Office-Paket des US-amerikanischen Unternehmens Microsoft für die Betriebssysteme Microsoft Windows und Mac OS X. Für unterschiedliche Aufgabenstellungen werden verschiedene Suites angeboten, die sich in den enthaltenen Komponenten, dem Preis und der Lizenzierung unterscheiden. Die Firma Microsoft veröffentlichte unlängst ein Advisory, indem er eine Schwachstelle (Pufferüberlauf) in verschiedenen Versionen des Produktes beschreibt. Diese Schwäche erlaubt es einem Angreifer, Kontroller über das System zu erlangen und seine Rechte zu erweitern.

#### Expertenmeinung:

Die vorliegende Schwachstelle ist zwar nicht hochkritisch, sollte aber aufgrund der hohen Verbreitung von Microsoft Office möglichst zeitnah Aufmerksamkeit erhalten. Das Einspielen der entsprechenden Patches wird aus diesen Gründen stark empfohlen.

### 3.9 Apple Safari verschiedene Schwachstellen

Risiko: **kritisch**  
 Remote: Ja  
 Datum: 08.06.2010  
 scip DB: <http://www.scip.ch/?vuldb.4132>

Safari ist ein Webbrowser des Unternehmens Apple für das hauseigene Betriebssystem Mac OS X und seit dem 11. Juni 2007 auch für Microsoft Windows, zunächst als Betaversion und seit der Versionsnummer 3.1 als stabile Version, erhältlich. Safari gehört zum Lieferumfang von Mac OS X ab der Version 10.3 ("Panther") und ersetzte den vorher mitgelieferten Microsoft Internet Explorer für Mac als Standard-Browser. Apple beschreibt in einem Advisory eine Schwachstelle (Pufferüberlauf) in aktuellen Versionen der Applikation. Durch die Ausnutzung der vorliegenden Schwachstelle kann unter

Umständen beliebiger Code zur Ausführung gebracht werden, was zur Kompromittierung des Systems führen kann.

**Expertenmeinung:**

Die diversen Schwachstellen, die Apple im vorliegenden Advisory beschreibt sind durchgehend als kritisch zu betrachten und sollten zeitnah durch das Einspielen entsprechender Patches mitigiert werden.

## 4. Statistiken Verletzbarkeiten

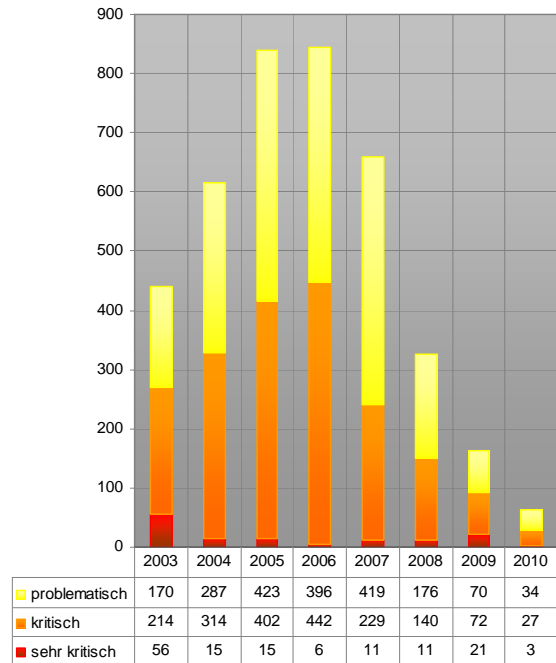
Die im Anschluss aufgeführten Statistiken basieren auf den Daten der deutschsprachige Verletzbarkeitsdatenbank der scip AG.



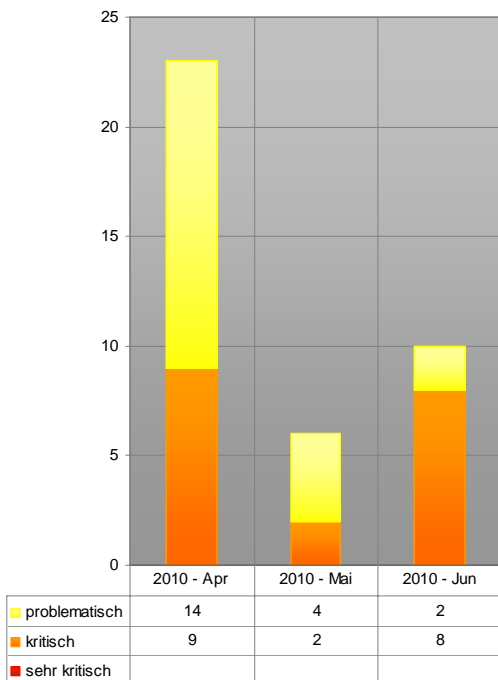
<http://www.scip.ch/?vuldb>

Zögern Sie nicht uns zu kontaktieren. Falls Sie spezifische Statistiken aus unserer Verletzbarkeitsdatenbank wünschen so senden Sie uns eine E-Mail an <mailto:info@scip.ch>. Gerne nehmen wir Ihre Vorschläge entgegen.

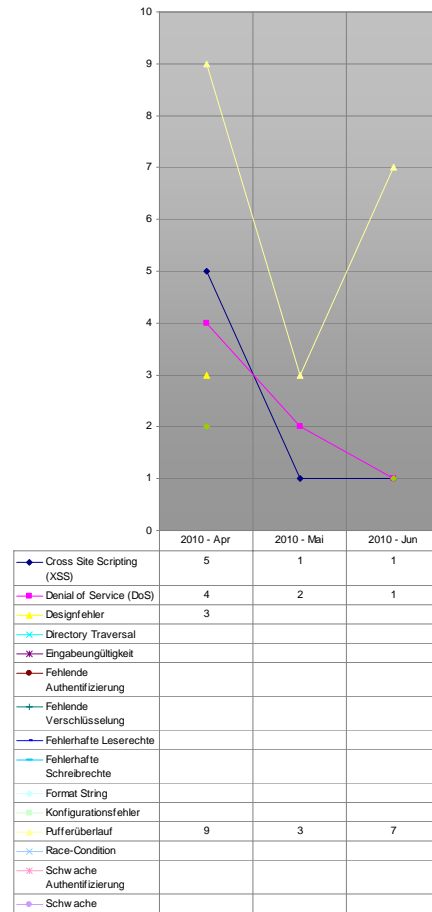
Auswertungsdatum: 19. Juni 2010



Verlauf der Anzahl Schwachstellen pro Jahr

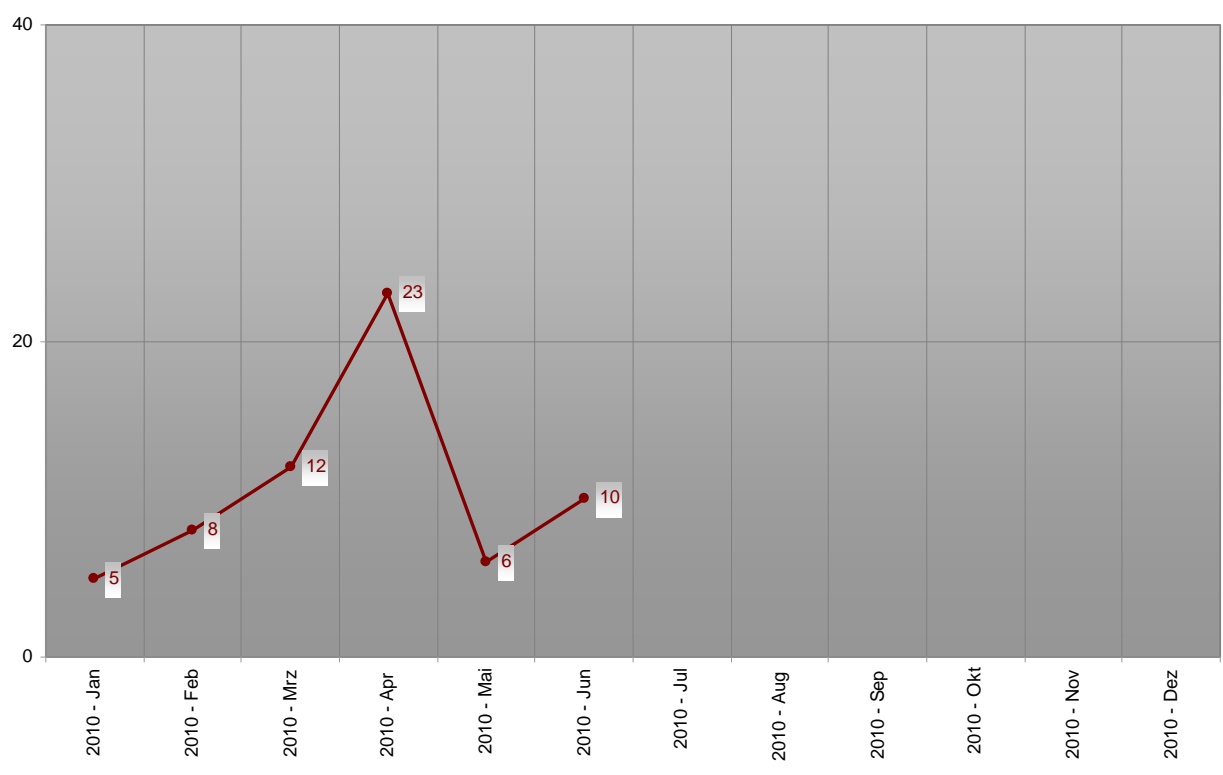


Verlauf der letzten drei Monate Schwachstelle/Schweregrad

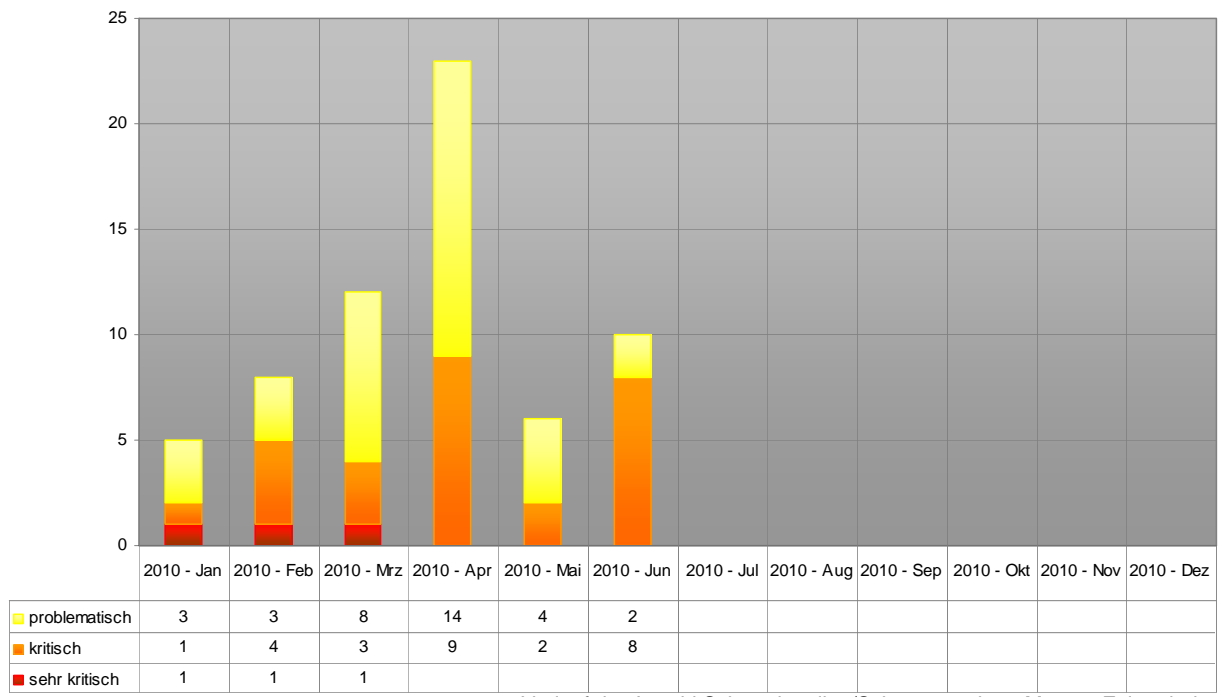


Verlauf der letzten drei Monate Schwachstelle/Kategorie

Registrierte Schwachstellen by scip AG



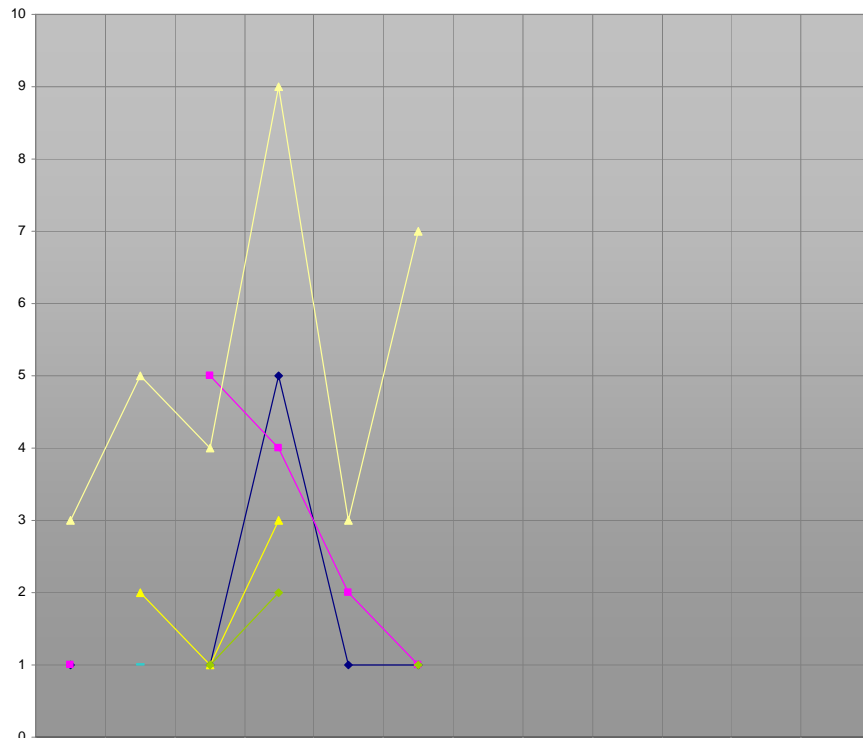
Verlauf der Anzahl Schwachstellen pro Monat - Zeitperiode 2010



Verlauf der Anzahl Schwachstellen/Schweregrad pro Monat - Zeitperiode 2010

scip monthly Security Summary 19.06.2010

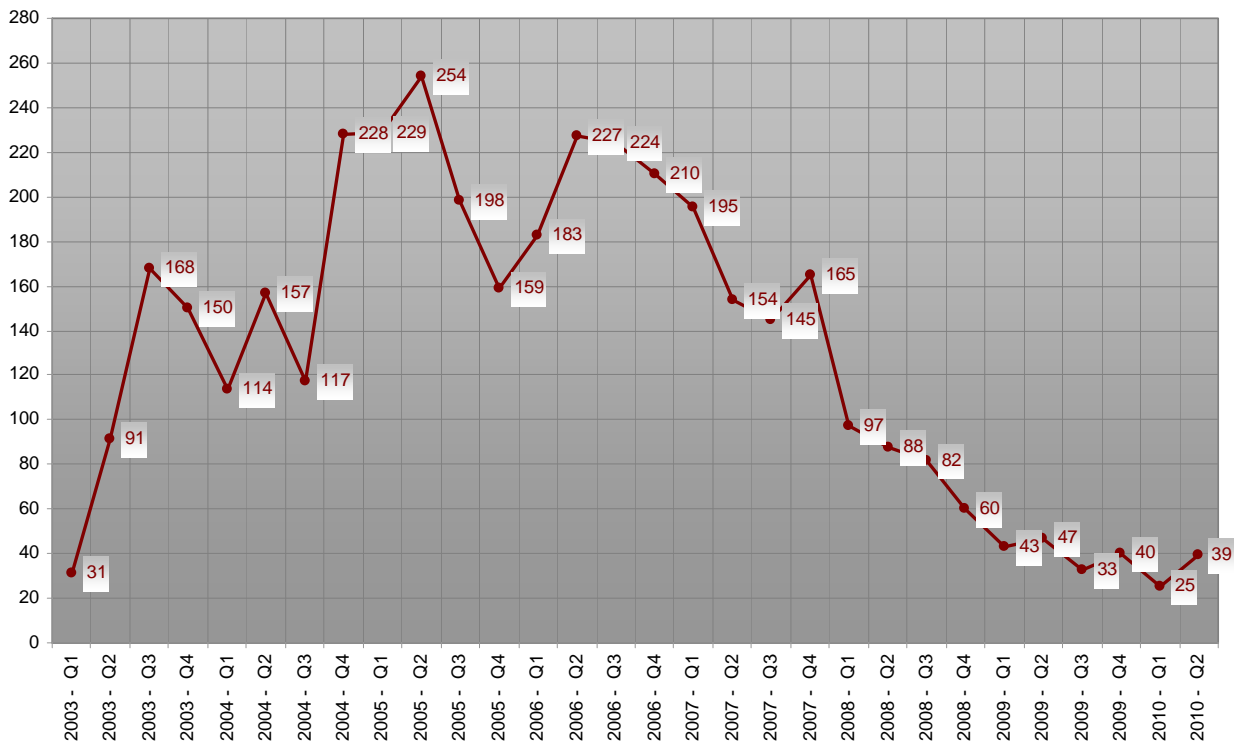




	2010 - Jan	2010 - Feb	2010 - Mrz	2010 - Apr	2010 - Mai	2010 - Jun	2010 - Jul	2010 - Aug	2010 - Sep	2010 - Okt	2010 - Nov	2010 - Dez
◆ Cross Site Scripting (XSS)	1		1	5	1	1						
◆ Denial of Service (DoS)	1		5	4	2	1						
◆ Designfehler		2	1	3								
◆ Directory Traversal												
◆ Eingabeungültigkeit												
◆ Fehlende Authentifizierung												
◆ Fehlende Verschlüsselung												
◆ Fehlerhafte Leserechte												
◆ Fehlerhafte Schreibrechte												
◆ Format String												
◆ Konfigurationsfehler												
◆ Pufferüberlauf	3	5	4	9	3	7						
◆ Race-Condition												
◆ Schwache Authentifizierung												
◆ Schwache Verschlüsselung												
◆ SQL-Injection												
◆ Symink-Schwachstelle												
◆ Umgehungs-Angriff		1										
◆ Unbekannt			1	2		1						

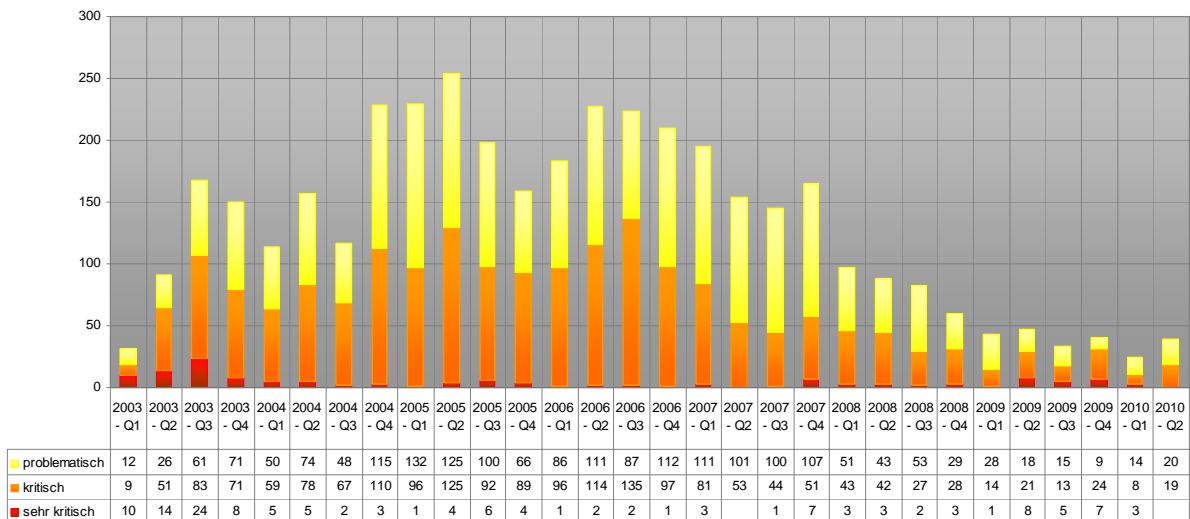
Verlauf der Anzahl Schwachstellen/Kategorie pro Monat - Zeitperiode 2010

Registrierte Schwachstellen by scip AG



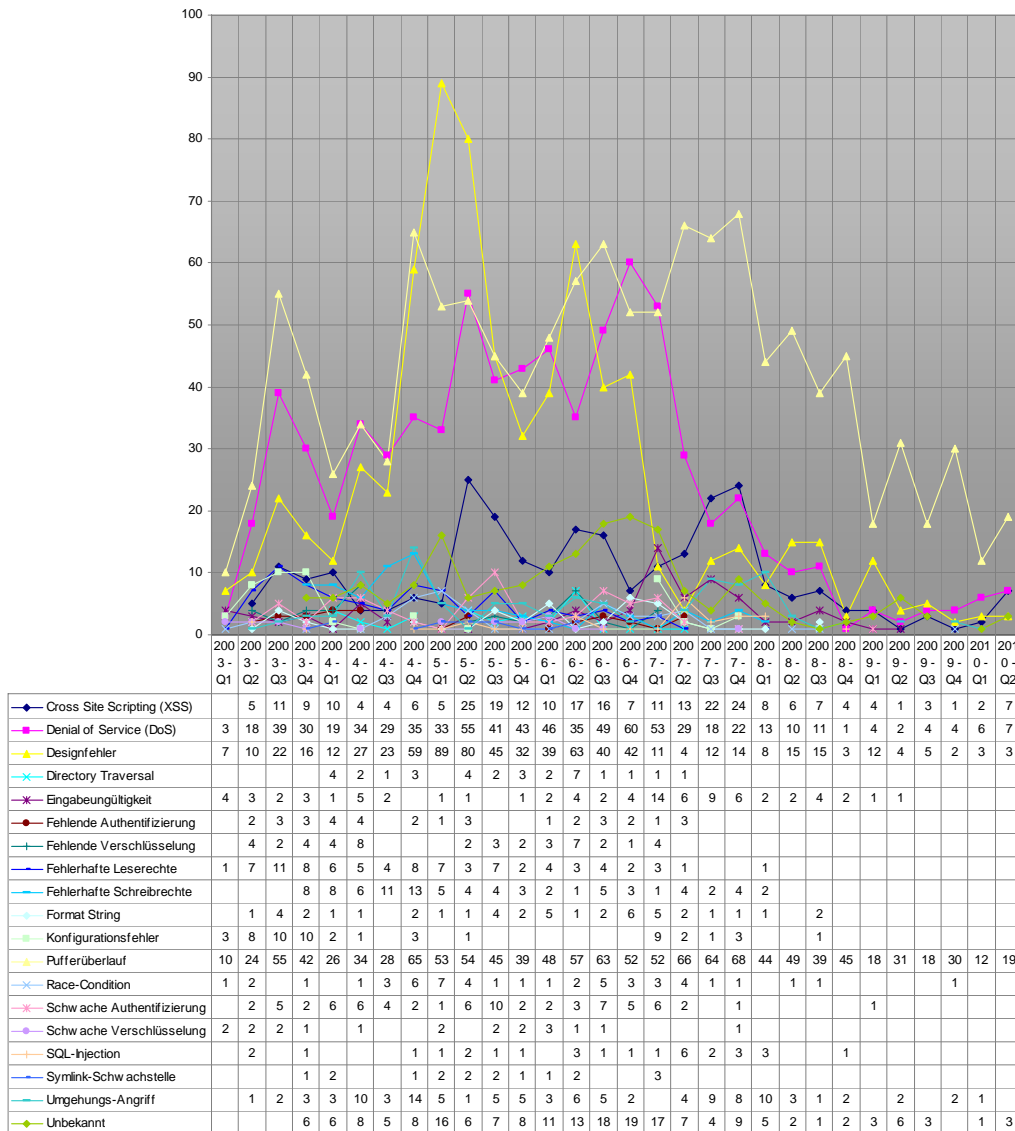
Verlauf der Anzahl Schwachstellen pro Quartal seit 2003 Q1

scip monthly Security Summary 19.06.2010



Verlauf der Anzahl Schwachstellen/Schweregrad pro Quartal seit 2003 Q1





Verlauf der Anzahl Schwachstellen/Kategorie pro Quartal seit 2003 Q1

## 5. Labs

Auf der scip Labs Webseite (<http://www.scip.ch/?labs.archiv>) werden regelmässig Neuigkeiten und Forschungsberichte veröffentlicht.

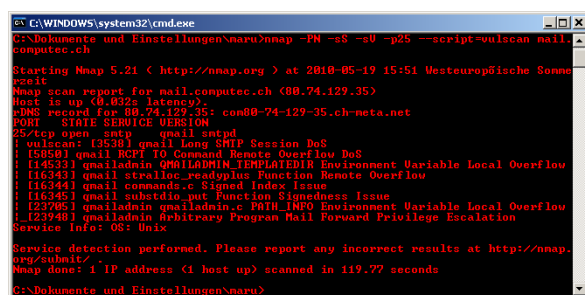
### 5.1 Nmap NSE Vulscan Script

03.06.2010 Marc Ruef, [maru@scip.ch](mailto:maru@scip.ch)

Das quelloffene Netzwerkutility [nmap](#) kann dank der [Nmap Scripting Engine](#) um eigene Mechanismen erweitert werden. Wir haben vor einigen Wochen eine 7-teilige Serie mit dem Titel [Nmap NSE Hacking](#), sie führt in das Thema von NSE-Skripting auf der Basis von Lua ein, veröffentlicht. Im Zuge dieser Publikation haben wir ebenfalls eine NSE-Portierung von [httprecon](#), einem Tool zur Umsetzung von [HTTP-Fingerprinting](#), umgesetzt.

Da wir einen Grossteil unserer [netzwerkbasieren Sicherheitsüberprüfungen](#) mit der Hilfe von nmap durchführen, schreiben wir stetig neue NSE-Skripte. Eine der grössten Entwicklungen in diesem Bereich ist das [nmap NSE Vulscan script](#). Dieses erweitert nmap, das ursprünglich als Portscanner konzipiert wurde, um die Funktionalität eines Vulnerability Scanners. Damit können im weitesten Sinn die Möglichkeiten geboten werden, wie sie zum Beispiel mit Lösungen wie [Nessus](#) oder [Qualys](#) genutzt werden können – *Nämlich das Erkennen von potentiellen Schwachstellen*.

Das nmap NSE Vulscan Script steht [hier](#) zum Download zur Verfügung.



```

C:\WINDOWS\system32\cmd.exe
C:\Dokumente und Einstellungen\maru\ntmap -PN -sS -sU -p25 --script=vulscan mail.computec.ch
Starting Nmap 5.21 ( http://nmap.org ) at 2010-06-19 15:51 Westeuropäische Sonne
nmap
Nmap scan report for mail.computec.ch (80.74.129.35)
Host is up (0.032s latency).
DNS record for 80.74.129.35: con80-74-129-35.ch-meta.net
PORT      STATE SERVICE VERSION
25/tcp    open  smtp      qmail-smtpd
| vulscan (12398) qmail Long SMTP Session DoS
| (15958) qmail RCPT TO Command Remote Overflow DoS
| (14933) qmailadmin QMAILADMIN_TEMPLATEDIR Environment Variable Local Overflow
| (16343) qmail stables_readopibus Function Remote Overflow
| (16344) qmail commands.c Signed Index Issue
| (16345) qmail substdio_put Function Signedness Issue
| (12398) qmailadmin qmailadmin.c PATH_INFO Environment Variable Local Overflow
| (12398) qmailadmin Arbitrary Program Mail Forward Privilege Escalation
Service Info: OS: Unix
Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 119.77 seconds
C:\Dokumente und Einstellungen\maru>

```

Wir hatten verschiedene Ziele vor Augen, als wir das Vulscan-Skript entwickelt haben. Als erstes ging es darum direkten Nutzen aus [Version Detection](#), eine sehr effiziente Implementierung des *Application Fingerprinting* durch nmap, zu ziehen. So versucht nmap auf der Basis eines Reiz/Reaktion-Schemas das angebotene Anwendungsprotokoll sowie im gleichen Zug die eingesetzte Server-Software (Hersteller, Produkt, Name, Version und zusätzliche Informationen) zu ermitteln. Diese Daten werden genutzt, um dar-

aus die potentiell vorhandenen Schwachstellen in der gegenwärtigen Installation auszumachen.

Zu diesem Zweck werden die durch nmap zur Verfügung gestellten Grunddaten weiterverwendet und mit [osvdb.org](#) verglichen. Hierbei handelt es sich um eine quelloffene Verwundbarkeitsdatenbank – ähnlich wie unsere [VulIDB](#). Dabei unterstützt die gegebene Implementierung [zwei unterschiedliche Lookup-Prozeduren](#):

- **Title Search:** Die OSVDB ist darum bemüht, *im Titel* einer Schwachstelle in stetig gleicher Weise das betroffene Produkt zu nennen. Die Volltextsuche macht sich diese Eigenschaft zu nutze und kann deshalb sehr schnell die möglichen Schwachstellen finden.
- **Correlations Lookup:** In der OSVDB werden ebenfalls Hersteller/Produkte/Versionen geführt, die ihrerseits mit den jeweiligen Schwachstellen *verknüpft* werden. Durch das Erkennen dieser verlinkten Einträge wird es möglich, sehr exakt die jeweiligen Schwachstellen zu bestimmen.

Standardmässig wird die *Title Search* umgesetzt. Sie ist sehr effizient, da sie lediglich ein Textfeld der Tabelle `vulnerabilities` durchsuchen muss. Jedoch kann dieser Modus verhältnismässig viele *False-Positives* generieren. Durch eine spezielle [Fuzzy Search](#) wird versucht die besten Treffer zu bestimmen. Da OSVDB und nmap jedoch nicht immer die gleichen Produktennamen verwenden, kann es hierbei zu Unstimmigkeiten kommen. Eine Vielzahl von fehlerhaften Treffern wird beispielsweise bei einem Apache-Webserver generiert.

Bessere Zuverlässigkeit und damit weniger False-Positives bietet der *Correlations Lookup*. Dieser Modus wird aktiviert, wenn beim Aufruf von nmap der Parameter `--script-args vulscan-correlation=1` verwendet wird. Sodann wird in einem ersten Schritt in der Tabelle `products` das Produkt bestimmt, um danach über die jeweiligen Zwischentabellen die verknüpften Verwundbarkeiten in der Tabelle `vulnerabilities` auszumachen. Dieser Prozess ist sehr aufwändig, da die Verknüpfungen berücksichtigt und deshalb verschiedene Tabellen durchsucht werden müssen. Da die Betreuer der OSVDB jedoch [nicht](#) alle Schwachstellen mit den jeweils verwundbaren Produkten verknüpft haben, neigt dieser Modus zu False-Negatives.

Wir arbeiten an verschiedenen Massnahmen, die aufgezeigten Schwächen der bereitgestellten Methoden [zu eliminieren](#). Das Vulscan-Skript hilft

jedoch sehr gut dabei, potentielle Schwachstellen in bekannten Produkten ausmachen zu können. Damit wird eine einfache und effiziente Grundlage geschaffen, um ein breitflächiges [Vulnerability Assessment](#) voranzutreiben und damit einen zielgerichteten [Penetration Test](#) angehen zu lassen.

Update 09.06.2010: Nach der Veröffentlichung des Skripts wurde durch Fyodor, der Lead Architect von nmap, die Diskussion [angestossen](#), die erweiterte Funktionalität in den Kern des Projekts zu übernehmen

Anyway, thanks for starting this exciting project and I hope Nmap proper will have this functionality someday.

## 6. Bilderrätsel



GESUCHTE BEGRIFFE		
8 (english)	12 (english)	10 (english)

LÖSUNGSWORT

### Wettbewerb

Mailen Sie uns das Lösungswort an die Adresse <mailto:info@scip.ch> inklusive Ihren Kontakt-Koordinaten.

Das Los entscheidet über die Vergabe des Preises. Teilnahmeberechtigt sind alle ausser den Mitarbeiterinnen und Mitarbeitern der scip AG sowie deren Angehörige. Es wird keine Korrespondenz geführt. Der Rechtsweg ist ausgeschlossen.

Einsendeschluss ist der **15.07.2010**.

Die GewinnerInnen werden nur auf ihren ausdrücklichen Wunsch publiziert. Die scip AG übernimmt keinerlei, wie auch immer geartete Haftung im Zusammenhang mit irgendeinem im Rahmen des Gewinnspiels an eine Person vergebenen Preises.

Gewinnen Sie ein Exemplar des Buches „Die Kunst des Penetration Testing“ von Marc Ruef. Dem meistverkauften deutschsprachigen Penetration Testing Fachbuch auf dem Markt.



<http://www.computec.ch/mruef/?s=dkdpt>  
911 Buchseiten, ISBN 3-936546-49-5

## 7. Impressum

Herausgeber:



scip AG  
Badenerstrasse 551  
CH-8048 Zürich  
T +41 44 404 13 13  
<mailto:info@scip.ch>  
<http://www.scip.ch>

Zuständige Person:



Marc Ruff  
Security Consultant  
T +41 44 404 13 13  
<mailto:maru@scip.ch>

scip AG ist eine unabhängige Aktiengesellschaft mit Sitz in Zürich. Seit der Gründung im September 2002 fokussiert sich die scip AG auf Dienstleistungen im Bereich Information Security. Unsere Kernkompetenz liegt dabei in der Überprüfung der implementierten Sicherheitsmassnahmen mittels **Penetration Tests** und **Security Audits** und der Sicherstellung zur Nachvollziehbarkeit möglicher Eingriffsversuche und Attacken (**Log-Management** und **Forensische Analysen**). Vor dem Zusammenschluss unseres spezialisierten Teams waren die meisten Mitarbeiter mit der Implementierung von Sicherheitsinfrastrukturen beschäftigt. So verfügen wir über eine Reihe von Zertifizierungen (Solaris, Linux, Checkpoint, ISS, Cisco, Okena, Finjan, Trend-Micro, Symantec etc.), welche den Grundstein für unsere Projekte bilden. Das Grundwissen vervollständigen unsere Mitarbeiter durch ihre ausgeprägten Programmierkenntnisse. Dieses Wissen äussert sich in selbst geschriebenen Routinen zur Ausnutzung gefundener Schwachstellen, dem Coding einer offenen Exploiting- und Scanning Software als auch der Programmierung eines eigenen Log-Management Frameworks. Den kleinsten Teil des Wissens über Penetration Test und Log-Management lernt man jedoch an Schulen – nur jahrelange Erfahrung kann ein lückenloses Aufdecken von Schwachstellen und die Nachvollziehbarkeit von Angriffsversuchen garantieren.

Einem **konstruktiv-kritischen Feedback** gegenüber sind wir nicht abgeneigt. Denn nur durch angeregten Ideenaustausch sind Verbesserungen möglich. Senden Sie Ihr Schreiben an [smss-feedback@scip.ch](mailto:smss-feedback@scip.ch). Das Errata (Verbesserungen, Berichtigungen, Änderungen) der scip monthly Security Summarys finden Sie online. Der Bezug des scip monthly Security Summary ist **kostenlos**. [Anmelden!](#) [Abmelden!](#)