
mapWOC-Handbuch

Release 1.1.2

30. 09. 2011

Inhaltsverzeichnis

1	Einführung	1
1.1	Gefahren auf Webseiten	1
1.2	Was ist mapWOC?	1
1.3	Komponenten von mapWOC	1
1.4	Funktionsweise von mapWOC	2
1.5	mapWOC im Cluster-Betrieb	2
2	mapWOC installieren und konfigurieren	3
2.1	Systemvoraussetzungen	3
2.2	Installation	3
2.3	Konfiguration eines Einzelsystems	4
3	mapWOC benutzen	7
3.1	Die Benutzeroberfläche	7
3.2	URL-Listen	8
3.3	Scans	9
3.4	Honey-Clients	12
3.5	Ergebnisse	15
3.6	Redirector	16
3.7	Benutzer	17
4	mapWOC-Cluster aufbauen	19
4.1	Konfiguration eines weiteren Knotens	19
5	mapWOC-Konfigurationsdateien	21
5.1	mapwoc-master	21
5.2	mapwoc-node	23
5.3	mapwoc-redirector	24
5.4	Honey-Clients	25
5.5	Weitere relevante mapWOC-Dateien	27
6	CA für mapWOC aufsetzen	29
6.1	Hintergrund	29
6.2	Eine CA erstellen	29
6.3	Einbindung der Zertifikate	31
7	Impressum	33

Einführung

Dieses Handbuch beschreibt die Nutzung und technische Hintergründe von mapWOC Version 1.1.2.

1.1 Gefahren auf Webseiten

Webseiten werden zunehmend als Einfallstor für die Infektion der Rechner ihrer Besucher genutzt. Dabei haben die Betreiber ihren Webseiten nicht etwa selbst missbräuchliche Inhalte hinzugefügt. Vielmehr werden sie selbst zu Opfern von Angriffen, bei denen die Inhalte ihrer Seiten verfälscht wurden. Meist handelt es sich nur um ein kleines Iframe-Element, das in die Datenbank des Betreibers eingeschleust wurde und während der Generierung neuer Seiten dem Inhalt unbemerkt hinzugefügt wird.

Während der Darstellung solcher Seiten wird im Browser noch Inhalt von einem zweiten (i.d.R. nicht vertrauenswürdigen) Server nachgeladen. Dieser Inhalt ist dann böswillig und infiziert über Schwachstellen im Webbrowser den Rechner. Der vollständige Vorgang wird auch als Drive-By-Download bezeichnet.

1.2 Was ist mapWOC?

mapWOC dient zur automatisierten Überprüfung der Integrität von Webseiten und der Erkennung böswillig verfälschter Inhalte.

mapWOC steht für massive automated passive Web Observation Center:

- massive: umfassende virtuelle und native Browsersysteme, verwendbar als Einzel- oder hoch skalierbare Cluster-Lösung (bis zu 1 Mio URLs pro Tag pro Knoten)
- automated: automatisiertes Ansurfen von definierten eigenen URL-Listen, analysieren des Netzwerkverkehrs nach Schadsoftware
- passive: Verweilen für definierte Zeit auf jeder URL (Warten auf Angriff)

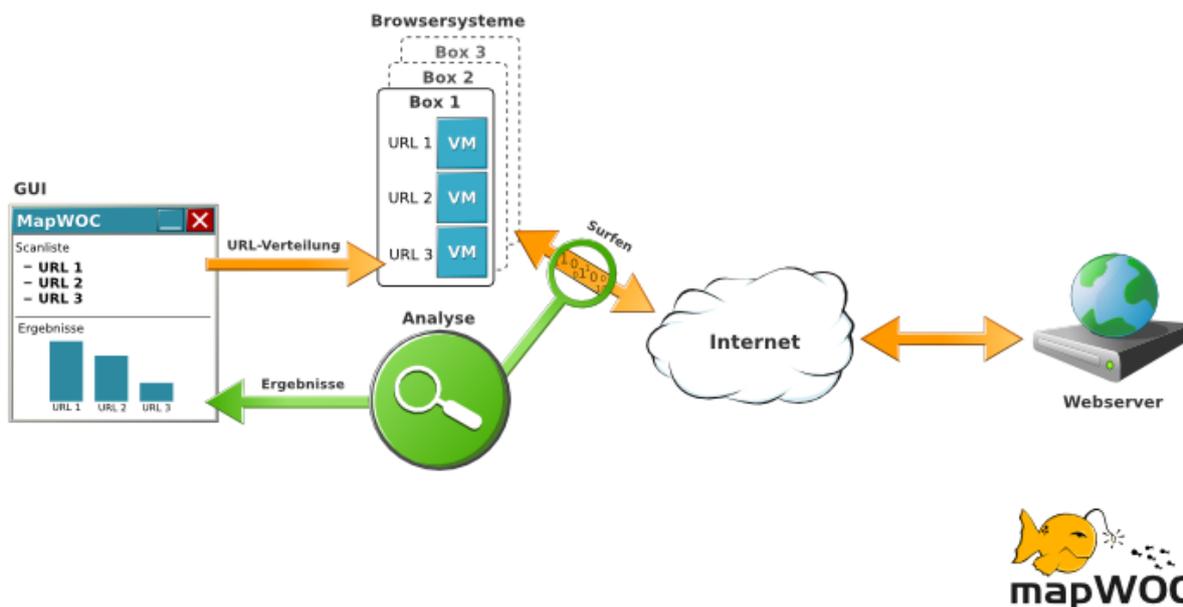
mapWOC wird vom [Bundesamt für Sicherheit in der Informationstechnik \(BSI\)](#) unterstützt.

1.3 Komponenten von mapWOC

mapWOC nutzt u.a. folgende Freie Software Komponenten:

- Debian Squeeze als Hostsystem
- KVM zur Virtualisierung
- ClamAV zur Untersuchung der Schadsoftware

1.4 Funktionsweise von mapWOC



1.5 mapWOC im Cluster-Betrieb

mapWOC kann in einem Rechnernetz (Cluster) betrieben werden, d.h. mapWOC kann auf mehr als einem Rechner laufen. Die Motivation für das Bilden eines Clusters ist die gesteigerte Performanz, also der Besuch und die Analyse von mehr URLs pro Zeit.

Ein Cluster besteht aus genau einem **Master** und beliebig vielen **Knoten**.

Der **Master** hält die Resultat-Datenbank und die Benutzeroberfläche; über letztere ist eine Interaktion mit den Knoten möglich.

Ein **Knoten** steuert die gewünschten Honey-Clients, analysiert den aufgezeichneten Netzwerkverkehr und überprüft gefundene Dateien auf Viren. Die Ergebnisse werden durch den Master über eine gesicherte Verbindung abgeholt und auf dem Knoten entfernt. Ein Knoten bietet keine Benutzeroberfläche an.

Es kann mehrere Knoten in einem Cluster geben, die allerdings alle eine eigene Verbindung zum Master benötigen. D.h. es können keine Knoten "hintereinander" verkettet werden. Der Master ist allen Knoten übergeordnet.

Auf einem physikalischen Rechner kann neben einem Master ebenfalls auch ein Knoten arbeiten. Eine typische Cluster-Konfiguration für mapWOC ist z.B.:

- Rechner 1: Master, Redirector + Knoten A
- Rechner 2: Knoten B
- Rechner 3: Knoten C
- ...

mapWOC installieren und konfigurieren

2.1 Systemvoraussetzungen

mapWOC nutzt extensiv folgende Technologien:

- `kvm`
- `python`
- `pylons`

Weiterhin wird eine Vielzahl an anderen freien Applikationen und Bibliotheken verwendet. Entwicklung und Tests finden auf Debian Squeeze-Systemen statt.

Zur Inbetriebnahme eines mapWOC-Systems werden zusätzlich KVM Gast-Images (= mapWOC-Honey-Clients) benötigt. Sofern Sie noch kein KVM-Image angelegt haben, befolgen Sie bitte vor dem ersten Starten der mapWOC-Dienste die Anleitung zur Erstellung eines virtuellen Honey-Clients im nächsten Kapitel mapWOC benutzen.

Außerdem müssen vor Inbetriebnahme von mapWOC einzelne Komponenten konfiguriert werden und das Host-System gegebenenfalls an bestehende Netzwerkkonfiguration angepasst werden.

Beachten Sie, dass die Erstellung der KVM Gast-Images und das Netzwerk-Setup keine trivialen Aufgaben sind. Bei Problemen wenden Sie sich an die mapWOC-Mailingliste (siehe www.mapwoc.org).

2.2 Installation

Die folgende Installationsanleitung bezieht sich auf Debian Squeeze – das empfohlene Hostsystem für mapWOC.

1. Ergänzen Sie `/etc/apt/sources.list` um folgende Zeile:

```
deb http://apt.intevation.org/ squeeze mapwoc
```

und optional (wenn Sie an den Quellpaketen interessiert sind):

```
deb-src http://apt.intevation.org/ squeeze mapwoc
```

2. mapWOC-Pakete installieren:

```
apt-get install mapwoc
```

Wichtig: Die bereitgestellten mapWOC-Pakete funktionieren nur für Debian Squeeze.

2.3 Konfiguration eines Einzelsystems

Bevor Sie mapWOC verwenden können, müssen Sie folgende Konfiguration an Ihrem Hostsystem und den mapWOC-Konfigurationsdateien vornehmen.

Bemerkung: Auf die Konfiguration eines gesicherten Verkehrs der Komponenten untereinander wird separat in im Abschnitt “*Setup CA*” eingegangen.

Bemerkung: Es wird hier eine Konfiguration zur Evaluation vorgestellt. Im Produktivbetrieb sollte mapWOC immer in Zusammenarbeit mit Sicherheitsbeauftragten und Systemadministratoren installiert werden.

Die drei Komponenten *mapwoc-master*, *mapwoc-node* und *mapwoc-redirector* lassen sich über Konfigurationsdateien oder über die Befehle parametrisieren. Eine ausführliche englischsprachige Beschreibung lässt sich mit

```
mapwoc-node --help
```

oder

```
man mapwoc-node
man mapwoc-node.config
```

aufrufen.

Als Voreinstellung wird die Datei `/etc/mapwoc/mapwoc-node.config` geladen. Es kann jedoch auch eine andere Konfigurationsdatei mit

```
mapwoc-node --config /path/to/testing-node.config
```

geladen werden. Das hier beispielhaft an *mapwoc-node* gezeigte Verhalten findet sich auch in *mapwoc-master* und *mapwoc-redirector*. (Vgl. auch Abschnitt “*mapWOC-Konfigurationsdateien*”.)

1. Neuen Nutzer anlegen:

```
adduser --disabled-password mapwoc
usermod -a -G kvm mapwoc
```

2. Umgebung anlegen:

```
mkdir -p /var/lib/mapwoc/master/credentials /var/lib/mapwoc/master/nodes/
mkdir -p /var/lib/mapwoc/master/work/
mkdir -p /var/lib/mapwoc/node/work /var/lib/mapwoc/node/results
chown -R mapwoc:mapwoc /var/lib/mapwoc
chmod 700 -R /var/lib/mapwoc/
```

(Wenn die Standard-Einstellungen bezüglich der Datenspeicherung benutzt werden sollen.)

Wichtig: Von nun an werden alle Aktionen als Nutzer *mapwoc* ausgeführt (*su - mapwoc*). Die Default-Konfigurationsdateien unter `/etc/mapwoc` müssen bei Bedarf für diesen Nutzer schreibbar gemacht werden.

3. Sich bekannt machen

Der *mapwoc-master* wird in regelmäßigen Abständen mit dem *rsync*-Programm nach neuen Ergebnissen auf den Nodes sehen. Diese Art der Kommunikation ist mit *ssh*-Schlüsseln abgesichert. Ein solches Schlüssel-paar muss einmalig für den Master erstellt werden und der öffentliche Teil auf jeder Node als vertrau-

enswürdig hinterlegt werden. Weiterhin müssen dem master die hosts auf denen die Nodes laufen bekannt gemacht werden.

```
:: mkdir /home/mapwoc/.ssh chmod 700 /home/mapwoc/.ssh
```

Das Schlüsselpaar wird (passwortlos) mit:

```
ssh-keygen -f master_key
```

erzeugt und (entsprechend der Konfigurationsdatei) in `/var/lib/mapwoc/master/key/` hinterlegt (*Achtung*: restriktive 0700-Rechte für diesen Ordner mit `chmod 700 <dir>` setzen, ansonsten wird das Schlüsselpaar nicht benutzt!):

```
mv /home/mapwoc/master_key* /var/lib/mapwoc/master/credentials/
chmod 700 /var/lib/mapwoc/master/credentials/
```

Nun wird ein Remote-Login mit diesem Schlüssel erlaubt:

```
cat /var/lib/mapwoc/master/credentials/master_key.pub >> ~/.ssh/authorized_keys
```

Und der Host wird bekanntgemacht:

```
touch ~/.ssh/known_hosts
echo -n "localhost ssh-rsa " >> ~/.ssh/known_hosts
ssh-keygen -e -f /etc/ssh/ssh_host_rsa_key | tail -n+3 | head -n-1 |
    awk '{printf "%s", $0}' >> ~/.ssh/known_hosts
```

Alternativ zu letztem Schritt kann:

```
ssh localhost
```

aufgerufen und die dann aufkommende Frage mit 'yes' beantwortet werden (danach mit Ctrl-C abbrechen).

4. Honey-Client-Dateien für die Node hinterlegen.

Dazu müssen Sie vorher Honey-Clients angelegt haben. Informationen dazu finden Sie im nächsten Kapitel mapWOC benutzen.

5. Datei-Rechte oder Pfade in der Datei `/etc/mapwoc/mapwoc-node.config` anpassen.
6. Datei-Rechte oder Pfade in der Datei `/etc/mapwoc/mapwoc-master.config` anpassen.
7. Datei-Rechte oder Pfade in der Datei `/etc/mapwoc/mapwoc-redirector.config` anpassen.
8. Den `mapwoc-redirector` starten:

```
mapwoc-redirector
```

Bemerkung: Die drei Komponenten schreiben Log-Informationen auf die Standardausgabe `stdout` und in eine Komponenten-abhängige Logdatei (`mapwoc-node.log`, `mapwoc-master.log`, `mapwoc-redirector.log`).

9. Eine `mapwoc-node` starten:

```
mapwoc-node
```

10. Dem `mapwoc-master` Zugang zu der `mapwoc-node` bekannt geben:

Dazu im **node_config_dir** (über die Konfigurationsdatei `/etc/mapwoc/mapwoc-master.config` auf `/var/lib/mapwoc/master/nodes/` voreingestellt), eine Datei anlegen, z.B. `demo-node.config`:

```
[demo-node]
name=Demo Node
description=This node runs on the same host as the master
rsync_address=localhost
```

```
ip=localhost
port=18158
```

11. *mapwoc-master* starten:

```
mapwoc-master
```

12. Die grafische Oberfläche aufsetzen und starten:

- Eine Ausgangs-Konfigurationsdatei erstellen.

```
paster make-config "mapwocgui" mapwoc-gui.ini
```

- Die erstellte Datei `mapwoc-gui.ini` gegebenenfalls anpassen:

Soll ein bereits existierendes Zertifikat genutzt werden, muss die Zeile

```
ssl_pem =
```

durch

```
ssl_pem = /path/to/certificate
```

ersetzt werden. Soll mit selbst-signierten Zertifikaten gearbeitet werden, wird

```
ssl_pem = *
```

eingetragen. Soll die ssl-Verschlüsselung ausgeschaltet werden, kann die Zeile entfernt oder durch ein “#” auskommentiert werden. Weiterhin kann der Port und das Interface konfiguriert werden. Mit

```
host = 0.0.0.0
port = 8080
```

wird die grafische Oberfläche über Port 8080 auf allen Netzwerkinterfaces zur Verfügung stehen. Außerdem sollten die Werte für **beaker.session.secret** und **authkit.cookie.secret** durch mehr oder weniger zufällige Werte ersetzt werden. Wurde *mapwoc-master* so konfiguriert, dass es auf einen anderen Port oder anderem Interface lauscht, ist der Wert **master.port** (**master.host**) anzupassen.

- Datenbank und initialen Nutzer einrichten:

```
paster setup-app mapwoc-gui.ini
```

- Die GUI starten:

```
paster serve mapwoc-gui.ini
```

mapWOC benutzen

3.1 Die Benutzeroberfläche

mapWOC ist über eine Web-Oberfläche steuerbar, die sich in ein Drei-Spalten-Layout gliedert (siehe Screenshot):

1. Hauptmenü
2. Inhaltsbereich
3. Login- und Mitteilungsbereich

The screenshot displays the mapWOC web interface. On the left is a sidebar menu with the mapWOC logo and icons for Status, URL-Listen, Scans, Ergebnisse, Honey-Clients, Knoten, Redirector, Benutzer, and Hilfe. The main content area is titled 'Systemüberblick' and contains three sections: 'Aktueller Scan' (Master ist: idle), 'Status vom Redirector' (Status: idle), and 'Status der Knoten' (wocepress: idle). The 'Status vom Redirector' section includes 'Letzte ausgegebene URLs' (none since the last URL list load) and 'Nächste URLs' (Redirector has no further URLs to output). On the right is a 'Meldungen' panel showing 'Noch keine Meldungen'. At the top right, the user 'admin Administrator' is logged in. The footer shows 'mapwoc © 2011, BSI'.

Aus Sicherheitsgründen verzichtet mapWOC vollständig auf JavaScript und andere aktive Inhalte.

In den nachfolgenden Abschnitten wird anhand der einzelnen Hauptmenüpunkte die Bedienung von mapWOC erläutert.

3.2 URL-Listen

Bei mapWOC konzentriert sich alles auf einzelne Webseiten (URLs). Ein Scan basiert auf einer vorher angelegten URL-Liste. URL-Listen bestehen aus beliebig vielen URLs.

Es gibt in mapWOC zwei Möglichkeiten URL-Listen zu **erstellen**:

1. URLs manuell eingeben

Vergeben Sie einen frei wählbaren Namen für die Liste. Der optionale Kommentar hilft beim späteren Zuordnen der Liste.

Wichtig beim Eintragen der URLs ist die richtige Schreibweise, in der Form: `http://www.example.com`. mapWOC überprüft die Validität aller URLs beim Erstellen. Sollte bei einer URL z.B. ein `http://` fehlen, wird diese URL aus der Liste entfernt. Der Nutzer bekommt eine Fehlermeldung. Die Liste wird angelegt - jedoch ohne diese fehlerhafte Adresse.

2. URL-Liste hochladen

URL-Listen lassen Sie auch aus vorliegenden Textdateien erstellen. Dazu die Datei auswählen und hochladen.

Achtung: Beachten Sie, dass beim Anlegen von sehr großen Listen das Hochladen, die Überprüfung und die Übertragung in die Datenbank einige Minuten dauern kann. Unterbrechen Sie diesen Prozess bitte nicht.

Im unteren Abschnitt der URL-Listen-Seite sind alle verfügbaren URL-Listen mit Anzahl der enthaltenen URLs aufgelistet. Ein Klick auf eine Liste zeigt weitere Details (Erstellungsdatum, URLs, Verknüpfung zu den Ergebnissen).

The screenshot displays the mapWOC interface for adding a new URL list. The sidebar on the left contains navigation icons for Status, URL-Listen, Scans, Ergebnisse, Honey-Clients, Knoten, Redirector, Benutzer, and Hilfe. The main content area is titled 'URL-Liste hinzufügen' and is divided into two sections: 'URLs manuell eingeben' and 'URL-Liste hochladen'. The 'manuell eingeben' section includes input fields for 'Name:', 'Kommentar:', and a large text area for 'URLs:', with a 'URL-Liste erstellen' button below. The 'hochladen' section includes input fields for 'Name:', 'Kommentar:', and a file selection field with a 'Browse...' button, and a 'Datei hochladen' button. Below these sections is a table titled 'Verfügbare URL-Listen' with columns 'Name' and '# URLs'. The table shows one entry: 'QuickScan URL-List' with 'mit 1 URLs'. On the right side, there is a 'Meldungen' panel showing 'Noch keine Meldungen' and an 'Archiv' button at the bottom.

Name	# URLs
QuickScan URL-List	mit 1 URLs

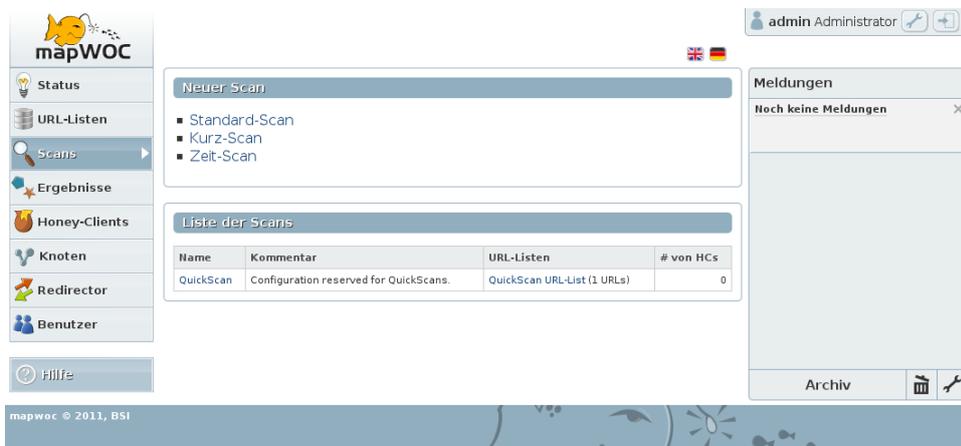
3.3 Scans

Ein Scan wird verwendet, um URLs in ausgewählten Browsern anzufordern und den Netzwerkverkehr zu analysieren.

3.3.1 Neuen Scan anlegen

mapWOC bietet drei Arten von Scans:

1. **Standard-Scan**
2. **Kurz-Scan**
3. **Zeit-Scan**



zu 1. **Standard-Scan**

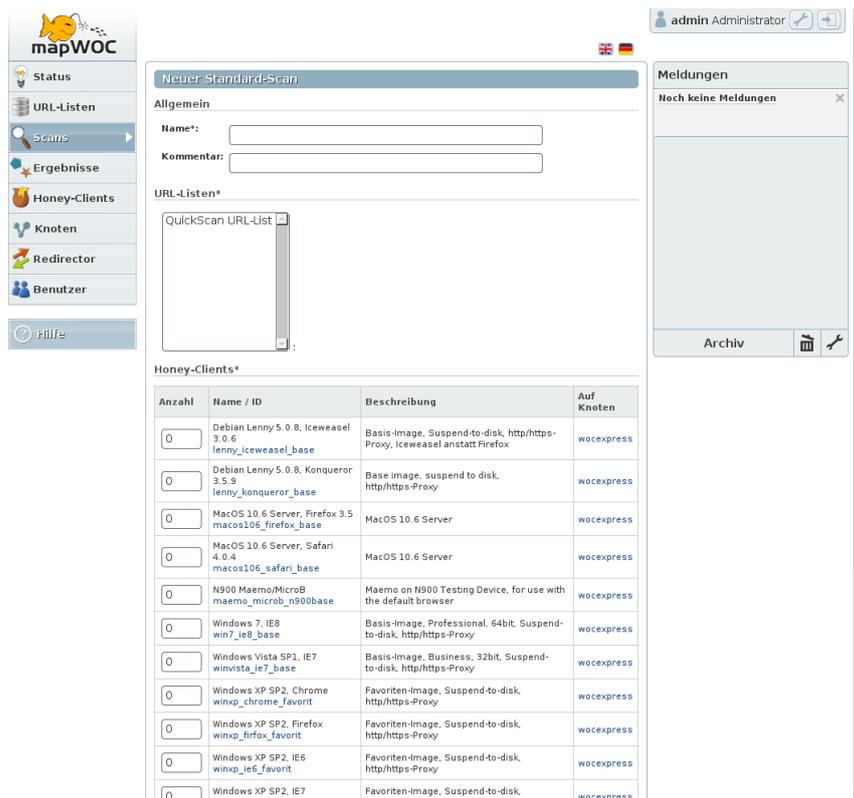
Ein Standard-Scan wird beendet, wenn die ausgewählten URL-Listen abgearbeitet sind.

Zum Anlegen eines Standard-Scans sind folgende Angaben möglich (siehe Screenshot):

- Name vergeben
- Kommentar vergeben (optional)
- URL-Liste(n) auswählen
- Honey-Clients auswählen (Die Anzahl der parallel laufenden HCs ist hier erforderlich. Der Wert ist abhängig von den verfügbaren Hardwareressourcen und Softwarelizenzen.)
- Mehrere Optionen sind auswählbar:
- *Aktives Zeitfenster*: Das ist die Zeit vom Starten bis zum Abbruch des Honey-Clients. Voreingestellt: 40 Sekunden.
- *Schnell-Modus*: Dabei werden gleich mehrere URLs pro Honey-Client aufgerufen (ohne *Schnell-Modus* wird nur eine URL pro HC verwendet). Der Browser holt sich vom Redirector mehrere URLs und öffnet diese parallel in ebenso vielen Browserfenstern. Auswählbar sind 2 bis 5 URLs pro Honey-Client. Dieser Modus ist voreingestellt deaktiviert.
- *Tasten-Ereignisse*: Nach einer bestimmten Zeit nach Start des virtuellen HCs kann das Ereignis 'Enter drücken' (e) oder 'Gehe zur Startseite/Homepage' (h) aufgerufen werden.

Ein Beispiel: Die Eingabe von 10e15h drückt automatisch nach 10 Sekunden Enter und nach weiteren 5 Sekunden (15 Sekunden nach Start) den Kurzbefehl für die Startseite (i.d.R. ALT+POS1).

Nach dem Anlegen eines Standard-Scans wird die Konfiguration auf einer Seite zusammengefasst. Zum Starten des Scans klicken Sie unter dem Abschnitt *Aktionen* auf *[Starten]*.



zu 2. Kurz-Scan

Ein Kurz-Scan wird beendet, nachdem die angegebene (Einzel-)URL angesurft wurde.

Zum Anlegen eines Kurz-Scans sind folgende Angaben erforderlich (siehe Screenshot):

- URL eingeben (in der Form `http://example.com`)
- Honey-Client auswählen

Zum Starten des Scans klicken Sie auf *[Scan starten]*.



zu 3. Zeit-Scan

Ein Zeit-Scan wird beendet, wenn das ausgewählte Zeitfenster abgelaufen ist. Ein Zeit-Scan dient zur manuellen Steuerung von nativen Honey-Clients. Virtuelle Honey-Clients können mit Zeit-Scans nicht betrieben werden.

Zum Anlegen eines Zeit-Scans sind folgende Angaben erforderlich (siehe Screenshot):

- nativen Honey-Client auswählen
- Zeitfenster festlegen (voreingestellt: 3 Minuten)

Zum Starten des Scans auf [Scan starten] klicken.

The screenshot displays the 'Neuer Zeit-Scan' configuration interface. On the left is a sidebar with navigation icons for Status, URL-Listen, Scans, Ergebnisse, Honey-Clients, Knoten, Redirector, and Benutzer. The main area is titled 'Neuer Zeit-Scan' and includes a 'QuickScan URL-List' dropdown menu. Below this is a table of 'Honey-Clients' with columns for Name / ID, Beschreibung, WLAN-SSID, and Auf Knoten. The 'Optionen' section features a dropdown for 'Aktives Zeitfenster' set to 3 Minuten and a 'Scan starten' button. A 'Meldungen' panel on the right indicates 'Noch keine Meldungen'.

Name / ID	Beschreibung	WLAN-SSID	Auf Knoten
N900 Maemo/MicroB maemo_microb_n900base	Maemo on N900 Testing Device, for use with the default browser	mw-101	wocexpress

3.3.2 Scan-Liste

Im unteren Abschnitt der Scan-Übersichtsseite sind alle bisher angelegten Scans gelistet. Dabei werden zu jedem Scan die verwendeten URL-Listen (mit Gesamtsumme aller URLs) sowie die verwendeten Honey-Clients (mit Gesamtsumme aller HCs) angezeigt. Der Kurz-Scan ist voreingestellt und enthält immer genau eine URL.

Ein Klick auf einen Scan zeigt alle Konfigurationsdetails an. Hier kann ein Scan gestartet, gestoppt oder gelöscht werden.

Wichtig: Das Bearbeiten eines Scans ist nicht möglich. Der Grund dafür ist, dass ein angelegter Scan mehrmals (hintereinander) durchgeführt werden kann. Um die Ergebnisse vergleichbar zu halten und eine eindeutige Referenzierung auf die Scankonfiguration zum Zeitpunkt des Scans zu gewährleisten, darf ein Scan nicht verändert werden. Möglich ist aber einen neuen Scan mit geänderter/ähnlicher Konfiguration anzulegen.

3.4 Honey-Clients

Ein Honey-Client (HC) ist ein Browsersystem, das zum Ansurfen von URLs in einem Scan genutzt wird.

Jedes beliebige Betriebssystem wird unterstützt, einzige Voraussetzung ist ein installierter Internetbrowser. mapWOC unterscheidet **virtuelle** und **native** Honey-Clients.

Der folgende Screenshot zeigt, wie so eine Liste mit Honey-Clients in mapWOC aussehen kann.

The screenshot shows the mapWOC web interface. On the left is a sidebar with navigation buttons: Status, URL-Listen, Scans, Ergebnisse, Honey-Clients (selected), Knoten, Redirector, Benutzer, and Hilfe. The main content area is titled 'admin Administrator' and shows two tables of Honey-Clients.

Liste der virtuellen Honey-Clients

Icon	hcsid	Name	Beschreibung	Auf Knoten
	lenny_iceweasel_base	Debian Lenny 5.0.8, Iceweasel 3.0.6	Basis-Image, Suspend-to-disk, http/https-Proxy, Iceweasel anstatt Firefox	wocexpress
	lenny_konqueror_base	Debian Lenny 5.0.8, Konqueror 3.5.9	Base image, suspend to disk, http/https-Proxy	wocexpress
	macos106_firefox_base	MacOS 10.6 Server, Firefox 3.5	MacOS 10.6 Server	wocexpress
	macos106_safari_base	MacOS 10.6 Server, Safari 4.0.4	MacOS 10.6 Server	wocexpress
	win7_ie8_base	Windows 7, IE8	Basis-Image, Professional, 64bit, Suspend-to-disk, http/https-Proxy	wocexpress
	winvista_ie7_base	Windows Vista SP1, IE7	Basis-Image, Business, 32bit, Suspend-to-disk, http/https-Proxy	wocexpress
	winxp_chrome_favorit	Windows XP SP2, Chrome	Favoriten-Image, Suspend-to-disk, http/https-Proxy	wocexpress
	winxp_firefox_favorit	Windows XP SP2, Firefox	Favoriten-Image, Suspend-to-disk, http/https-Proxy	wocexpress
	winxp_ie6_favorit	Windows XP SP2, IE6	Favoriten-Image, Suspend-to-disk, http/https-Proxy	wocexpress
	winxp_ie7_favorit	Windows XP SP2, IE7	Favoriten-Image, Suspend-to-disk, http/https-Proxy	wocexpress

Liste der nativen Honey-Clients

Icon	hcsid	Name	Beschreibung	Auf Knoten	SSID
	maemo_microb_n900base	N900 Maemo/MicroB	Maemo on N900 Testing Device, for use with the default browser	wocexpress	mw-101

At the bottom of the interface, there is a footer: mapwoc © 2011, BSI.

3.4.1 Virtuelle Honey-Clients

Virtuelle Honey-Clients sind Virtuelle Maschinen (VMn), die mit der Freien Virtualisierungslösung **KVM** erzeugt werden. Die Virtuellen Maschinen werden installiert, konfiguriert und mit einem gestarteten Browser abgespeichert (Snapshot) oder wahlweise in den Ruhezustand (suspend-to-disk) versetzt. Details zum Anlegen von KVM-Images für den Einsatz in mapWOC finden Sie im nächsten Abschnitt.

Der Snapshot/Ruhezustand eines virtuellen Honey-Clients wird von mapWOC geladen. Der Browser surft per Kurzbefehl die Redirector-Startseite an und holt sich eine URL ab. Nach Ablauf des aktiven Zeitfensters beendet mapWOC die VM wieder. Das KVM-Image ist anschließend wieder im Ausgangszustand (Snapshot/Ruhezustand). Änderungen werden nicht gespeichert.

Diese automatische Zurücksetzbarkeit ist der entscheidende Vorteil gegenüber nativen Systemen.

Honey-Clients sind über die Honey-Client-System-ID (hcsid) eindeutig bestimmt.

Virtuellen Honey-Client anlegen

Die folgende Anleitung (am Beispiel von Windows XP) hilft beim Anlegen eines KVM-Images, um es später in mapWOC als neuen virtuellen Honey-Client hinzuzufügen:

1. KVM-Master-Image erstellen (im qcow2-Format, mit einer 8 GB Festplatte):

```
kvm-create -f qcow2 winxp-master.img 8G
```

2. Installation des Betriebssystems durchführen:

```
kvm -cdrom windows-xp.iso -hda winxp-master.img -m 128M -vnc :1
```

Dabei auf einer anderen Konsole einen VNC-Viewer starten, z.B.:

```
vncviewer localhost:1
```

Installieren Sie alle erforderliche Software. Anschließend fahren Sie das Betriebssystem herunter.

4. KVM-Overlay-Image erstellen:

```
kvm-img create -b winxp-master.img -f qcow2 winxp_ie6_demo.ovl
```

5. KVM-Overlay-Image starten:

```
kvm -hda winxp_ie6_demo.ovl -m 128M -vnc :1
```

6. Browser starten (hier IE6).

7. Betriebssystem in den Ruhezustand versetzen.

Bei Windows XP: *Start > Beenden > Herunterfahren > Ruhezustand* (Umschalttaste drücken)

Das Betriebssystem speichert den Zustand in das Overlay-Image. Damit existiert nun ein Master- und ein Overlay-Image.

8. Damit mapWOC den neuen Honey-Client kennt, muss noch eine neue Konfigurationsdatei (hc-Datei) im hcsdir-Verzeichnis der Node angelegt werden.

Hier ein Beispiel für eine hc-Datei `winxp_ie6_demo.hc`:

```
[hc]
name=WindowsXP Demo Image IE6
description=Lots of other software on it
hcsid=winxp_ie6_demo
max_parallel=-1
controllable=true
active_browser=cpe://a:microsoft:ie6
installed_software=cpe:o:microsoft:etc

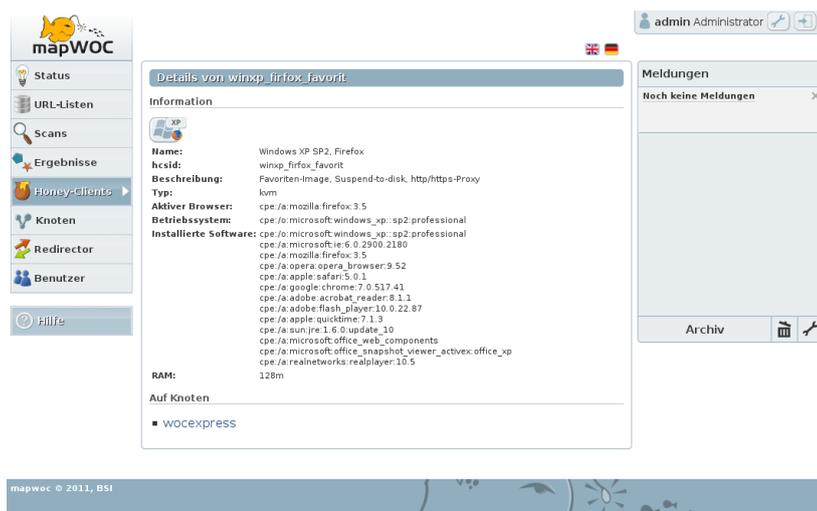
[control]
type=kvm
boots=true
image_master=/usr/share/mapwoc/winxp-master.img
image_file=/usr/share/mapwoc/images/winxp_ie6_demo.ovl
image_memory=128

[network]
own_ip=10.0.0.4
dns_ip=10.0.0.3
proxy_ip=10.0.0.2
gateway_ip=10.0.0.1
homepage=http://10.0.0.1/winxp_ie6_demo
```

9. Zuletzt muss nur noch die Node und der Master von mapWOC neu gestartet werden:

```
mapwoc-node --config=node.conf
mapwoc-master --config=master.conf
```

Der neue virtuelle Honey-Client `winxp_ie6_demo` erscheint nun in der Liste der Honey-Clients. Beim Klick auf diesen Eintrag lassen sich die Details anschauen (siehe Screenshot):



3.4.2 Native Honey-Clients

Native Honey-Clients sind Geräte, die nicht automatisiert von mapWOC gesteuert werden können (z.B. Mobilgeräte, Laptop). Automatisiert steuern kann mapWOC nur Virtuelle Maschinen (s.o.).

Native Honey-Clients bauen per WLAN eine Verbindung zu einem Access-Point auf, welcher am mapWOC-Server angeschlossen und eingerichtet ist.

Nativen Honey-Client hinzufügen

Um nun einen neuen nativen Honey-Client in mapWOC hinzuzufügen, befolgen Sie folgende Schritte:

1. MAC-Adresse des nativen Geräts über das Administrationsinterface des WLAN-Access-Points hinzufügen.
2. Eine neue hc-Konfigurationsdatei im `hcsdir`-Verzeichnis der Node anlegen.

Hier ein Beispiel für ein Nokia N900 mit Maemo und MicroB-Browser in der hc-Datei `maemo_microb_n900 hc`:

```
[hc]
name=N900, MicroB
description=Maemo
hcsid=maemo_microb_n900
max_parallel=1
controllable=false
active_browser=mapwoc-cpe:/a:nokia:microb
os=mapwoc-cpe:/o:nokia:maemo
installed_software=mapwoc-cpe:/o:nokia:maemo

[control]
type=native
interface=vlan103
image_memory=10

[network]
client_mac=c0:38:f9:ee:f0:ec
wlan_ssid=mw-103
proxy_ip=10.1.103.1:1066
```

3. Abschließend die Node und den Master von mapWOC neu starten:

```
mapwoc-node --config=node.conf
mapwoc-master --config=master.conf
```

Der neue native Honey-Client `maemo_microb_n900` erscheint danach in der Liste der Honey-Clients.

3.5 Ergebnisse

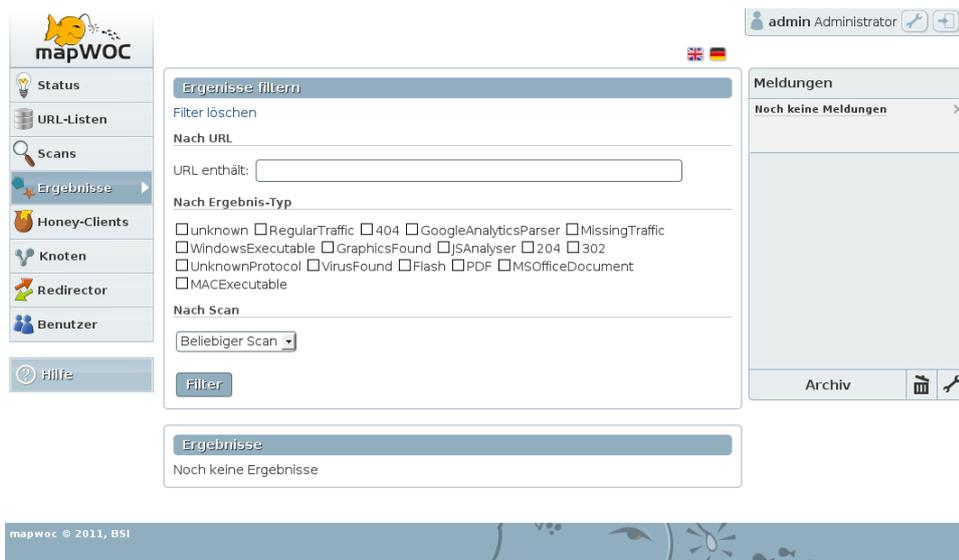
Die Ergebnisse aller Scans werden unter dem Menüpunkt *Ergebnisse* dargestellt.

Sie können Ergebnisse filtern:

- nach einer Zeichenkette in einer URL (z.B. die Domain `example.com`)
- nach einem oder mehreren Ergebnis-Typen
- nach einem verhangenen Scan

Alle Filter können Sie auch kombinieren.

Der folgende Screenshot zeigt Ihnen die (leere) Ergebnis-Übersichtsseite mit allen Filtermöglichkeiten:



3.5.1 Ergebnis-Typen

mapWOC analysiert den Netzwerkverkehr beim Besuch jeder URL. Dabei werden relevante, in einem Analyse-skript festgelegte Dateien aus dem sogenannten *pcap*-Strom extrahiert und mit einem integrierten Virenschanner (voreingestellt ist *ClamAV*) untersucht. Jede extrahierte Datei wird als Fund bezeichnet. Jeder Fund wird bewertet, wie sicherheitskritisch diese Datei sein könnte, und in einen Schweregrad von 0 bis 999 eingeteilt: 0 ist ungefährlich, 999 bedeutet sehr kritisch.

Die nachfolgende Auflistung zeigt die verfügbaren Ergebnis-Typen mit ihrem jeweiligen Schweregrad - sortiert in absteigender Reihenfolge:

- unknown (999)
- VirusFound (900)
- MissingTraffic (600)
- WindowsExecutable (500)
- GoogleAnalyticsParser (500)
- MACExecutable (211)
- MSOfficeDocument (210)
- Flash (200)
- PDF (199)
- 404 (100)

- JSAnalyser (100)
- 204 (99)
- 302 (60)
- GraphicsFound (50)
- UnknownProtocol (001)
- RegularTraffic (000)

3.6 Redirector

Der Redirector ist zuständig für die Verteilung der URLs.

In jedem Browsersystem muss die URL des Redirectors als Startseite eingestellt werden:

`http://10.0.0.1/<hcsid>`

Also z.B.: `http://10.0.0.1/winxp_ie6_demo`

Durch Aufrufen dieser URL bekommt das Browsersystem vom Redirector die nächste URL aus der Liste geliefert. Wird der *Schnell-Modus* verwendet, werden vom Redirector gleich n URLs zurückgeliefert, die in n Browserfenstern geöffnet werden. (n ist die Anzahl der Seiten pro Honey-Client).

Die Redirector-Seite bietet während eines Scans eine Übersicht über den Fortschritt der verwendeten URL-Listen. Bei den letzten abgerufenen URLs wird zusätzlich der Honey-Client und der Zeitpunkt angegeben. Die nächsten URLs geben einen Ausblick, welche URLs als nächstes vom Redirector verteilt werden.

The screenshot shows the mapWOC Redirector web interface. The interface is in German and displays the following information:

- Status:** Redirector Status: idle
- Fortschritt der URL-Liste:** 0,0% (0 von 0)
- Letzten 30 URLs:** Es wurden noch keine URLs ausgegeben.
- Nächsten 10 URLs:** 0 URLs remaining

The interface includes a sidebar with navigation options: Status, URL-Listen, Scans, Ergebnisse, Honey-Clients, Knoten, Redirector, Benutzer, and Hilfe. A top right corner shows the user 'admin Administrator' and a language selector. A bottom right corner has an 'Archiv' button and a trash icon.

3.7 Benutzer

mapWOC unterteilt Benutzer entsprechend ihres Aufgabenprofils in drei Rollen:

1. Gast

Ein Gast darf *keine* Änderungen vornehmen. Lediglich Informationen ansehen:

- Systemstatus, Ergebnisse, URL-Listen, Scankonfigurationen, Honey-Clients, Knoten, Redirector und Meldungen ansehen
- eigene Benutzereinstellungen ändern

2. Normaler Nutzer

Im Vergleich zum Gast darf der normale Nutzer zusätzlich:

- URL-Listen verwalten
- Scans anlegen/starten/stoppen

3. Administrator

Im Vergleich zum normalen Nutzer darf der Administrator zusätzlich:

- Benutzern verwalten
- neue Knoten hinzufügen/entfernen
- Systemeinstellungen ändern

The screenshot displays the mapWOC user interface. On the left is a sidebar with icons and labels for various system components: Status, URL-Listen, Scans, Ergebnisse, Honey-Clients, Knoten, Redirector, Benutzer, and Hilfe. The main area is titled 'Benutzer' and contains two primary configuration panels. The top panel, 'Passwort ändern für 'user'', features two text input fields for 'Neues Passwort' and 'Neues Passwort wiederholen', followed by a 'Passwort ändern' button. The bottom panel, 'Einstellungen ändern für 'user'', includes an 'E-Mail' input field, a 'Bevorzugte Sprache' dropdown menu currently set to 'Deutsch', and two checkboxes under 'Benachrichtigung senden' (one for 'bei Problemen' and one for 'bei abgeschlossenen Scans'). Below these is a 'Zugangsrechte' dropdown menu set to 'Gast' and an 'Einstellungen ändern' button. To the right of the main content is a 'Meldungen' (Notifications) panel with the message 'Noch keine Meldungen' and an 'Archiv' button at the bottom. The top right corner shows the user's profile as 'admin Administrator' with a language selector (UK, DE) and a user icon. The footer of the interface reads 'mapwoc © 2011, BSI'.

mapWOC-Cluster aufbauen

Zum Aufbau eines mapWOC-Clusters sind (mindestens) zwei Rechner notwendig. Beide Maschinen werden aufgesetzt und das mapWOC-System sowie die Abbilder der VMn installiert. Die beiden Rechner werden ans Netzwerk angeschlossen und entsprechend konfiguriert.

Ein Rechner *R1* wird zum Master *M* ernannt. Zusätzlich wird auf *R1* ein Knoten *K1* eingerichtet. Master und Knoten laufen in der Regel unter zwei unterschiedlichen Benutzern auf *R1*.

Auf Rechner *R2* wird ein Knoten *K2* eingerichtet.

Für *M* wird ein SSH-Schlüsselpaar erzeugt und auf den beiden Knoten *K1* und *K2* bekannt gemacht (*authorized_keys*). Ferner benötigt der Master zum autorisierten Zugriff auf die Knoten deren Fingerprints (*known_hosts*).

Über jeweils eine mapWOC-Konfigurationsdateien werden die beiden Knoten *K1* und *K2* auf dem Master registriert.

Nach dem Start aller mapWOC-Dienste sind *K1* und *K2* in der Benutzeroberfläche unter dem Menüpunkt *Knoten* sichtbar.

Um den Knoten *K2* aus dem Cluster herauszunehmen wird gegensätzlich gearbeitet: Der SSH-Schlüssel des Masters *A* wird auf *K2* entfernt. Ferner muss auf dem Master *M* die Konfigurationsdatei *K2* entfernt werden. Nach Neustart der mapWOC-Dienste (master und node) besitzt *M* nur noch einen Knoten *K1*.

4.1 Konfiguration eines weiteren Knotens

Achtung: Die hier angegebene Konfiguration ist als Beispiel zu verstehen und sollte für den Produktivbetrieb sorgfältig mit vorhandener Infrastruktur und Sicherheitspolicy in Einklang gebracht werden.

1. Pakete installieren:

```
apt-get install mapwoc
```

2. Nutzer einrichten:

```
adduser --disable-password mapwoc  
usermod -a -G kvm mapwoc
```

Wichtig: Folgende Schritte sind als Nutzer mapwoc auszuführen (sofern ausreichende Rechte vorhanden sind).

3. Zugriff auf Resultate via öffentlichen Schlüssel (default: `/var/lib/mapwoc/master/credentials/master_key.pub`) vom master erlauben:

```
mkdir ~/.ssh
# master_key.pub vom Master auf Node kopieren.
cat master_key.pub >> .ssh/authorized_keys
rm master_key.pub
```

4. Dem Master den Node-host bekannt machen.

Entweder `/home/mapwoc/.ssh/known_hosts` auf master-seite per Hand anpassen oder mit

```
ssh ip-of-node
```

SSH diesen Eintrag schreiben lassen (die aufkommende Frage ist dann mit 'yes' zu beantworten).

5. Eventuell `/etc/mapwoc/mapwoc-node.config` anpassen.

6. HC-Dateien für die node hinterlegen.

7. Auf der master-Seite ist eine node-configuration zu erstellen:

```
[remote-node]
name=Remote Node
description=This node runs on a different host than the master
rsync_address=hostname
ip=10.2.1.2
port=18158
```

(rsync_address und ip sind den Gegebenheiten anzupassen.)

1. Netzwerk

Es ist sicherzustellen, dass der Redirector-Port (i.d.R. 80) auf den master weitergeleitet wird (z.B. mit `ssh port-forwarding` oder `rinetd`).

2. Die Node starten:

```
mapwoc-node
```

Bemerkung: Auf den zusätzlichen Knoten müssen weder `mapwoc-redirector`, `mapwoc-master` noch die GUI gestartet werden. Es muss allerdings sichergestellt sein, dass die HoneyClients den Redirector auf dem master erreichen.

3. Den Master neu starten:

```
# Auf dem master
mapwoc-master
```

mapWOC-Konfigurationsdateien

In diesem Kapitel werden die wichtigsten Konfigurationsdateien von mapWOC erläutert.

5.1 mapwoc-master

Der *mapwoc-master* steuert eine oder mehrere *mapwoc-nodes* und hört auf einer Adresse auf Befehle und Anfragen (z.B. von der *mapwocgui*).

/etc/mapwoc/mapwoc-master.config ist die Konfigurationsdatei für den *mapwoc-master* Befehl.

Hier ein Beispiel für so eine Konfigurationsdatei:

```
[config]
name=Master
address=localhost
node_config_dir=/var/lib/mapwoc/master/nodes/
credentials_path=/var/lib/mapwoc/master/key/master_key
ctrl_port=8558
work_dir=/var/lib/mapwoc/master/work/
db=/var/lib/mapwoc/db
redirector_port=8228
tolerance=101
#sslconfig=/var/lib/mapwoc/master/credentials

[notifications]
# Configuration of Mail notifications.

# Server used to send Mails with
smtp_server=mail.domain
smtp_port=25

# Mail address to send administrative notifications to
admin_mail=admin.mapwoc-demo@domain

# Address used by mapWOC as sender
sender_mail=MapWOC Demo <demo@mapWOC>
```

Alle Werte aus der Konfigurationsdatei lassen sich auch als Option über den Befehl *mapwoc-master* setzen. Hier ein Überblick aller verfügbarer **Optionen** (Auszug aus der Manpage von *mapwoc-master*, siehe auch `mapwoc-master --help`):

```
-v, --verbose
    Be verbose and chatty on stdout and logfile.
```

`-q, --quiet`
Be quiet, only log warnings and errors.

`--redirector_port`
Port to talk to when talking to redirector.

`--ctrl_port`
Port to listen on for control commands.

`--name` Name of the master.

`--address`
Interface to listen on.

`--node_config_dir`
Directory in which to look for node config files.

`--credentials_path`
Path to credentials to use when syncing results with nodes.

`--work_dir`
Directory to place working data.

`--db` Path to the database.

`--tolerance`
Severity threshold (irrevertable).

`--sslconfig`
Path to directory containing the {ca,cert,key}.pem files used for secured communication (talk plain text if not given).

`--config`
Path to configuration file to read configuration from (or write to if called with `--generate-config`).

`--generate-config`
Generate a configuration file.

`-h, --help`
Show help and exit.

`--version`
Show version and exit.

5.2 mapwoc-node

Die *mapwoc-node* hört auf Befehle des *mapwoc-masters*. Typischerweise resultieren diese Befehle im Starten und Stoppen von KVM-Prozessen und Analysieren des Netzwerkverkehrs von den Honey-Clients.

/etc/mapwoc/mapwoc-node.config ist die Konfigurationsdatei für den *mapwoc-node* Befehl.

Hier ein Beispiel für so eine Konfigurationsdatei:

```
[config]
ctrl_port=18158
ctrl_interface=localhost
hcsdir=/var/lib/mapwoc/node/hcs
result_dir=/var/lib/mapwoc/node/results
work_dir=/var/lib/mapwoc/node/work
num_analyser=4
max_gb_ram=4
analyser=/usr/bin/mapwocanalyse.pl
native_gate_script=/usr/lib/mapwoc/dumper_native_hc.sh
proxy_script=/usr/lib/mapwoc/mapwoc_redirect_script

#Port Range the node may use to start https proxies
proxy_ports=10000-11000

#sslconfig=/var/lib/mapwoc/node/credentials
```

Alle Werte aus der Konfigurationsdatei lassen sich auch als Option über den Befehl *mapwoc-node* setzen. Hier ein Überblick aller verfügbarer **Optionen** (Auszug aus der Manpage von *mapwoc-node*, siehe auch `mapwoc-node --help`):

```
-v, --verbose
    Be verbose and chatty on stdout and logfile.

-q, --quiet
    Be quiet, only log warnings and errors.

--ctrl_port
    Port to listen on for control commands.

--hcsdir
    Directory from which to read HC files.

--work_dir
    Directory in which to work and store temporary
    files.

--result_dir
    Directory in which to place results that are fetched
    from master.

--num_analyser
    Number of analyser processes to start in parallel.

--max_gb_ram
    Number of analyser processes to start in parallel.

--analyser
    The program to execute for analysis.

--analyser_conf
    Path to configuration file for analyser.

--native_gate_script
```

```
The program to execute for control of native HCs.

--proxy_script
    Path to script for mitmproxy.

--proxy_ports
    The range of ports the node may use for ssl proxies

--sslconfig
    Path to credentials for secured communication.

--config
    Path to configuration file to read configuration from
    (or write to if called with --generate-config).

--generate-config
    Generate a configuration file.

-h, --help
    Show help and exit.

--version
    Show version and exit.
```

5.3 mapwoc-redirector

Der *mapwoc-redirector* startet zwei kleine Server. Auf dem einem Server werden HTTP-Anfragen mit HTTP-Redirects (Weiterleitungen) auf URLs einer Liste beantwortet. Der andere Server hört auf Kontrollbefehle, z.B. das Laden einer neuen URL-Liste.

/etc/mapwoc/mapwoc-redirector.config ist die Konfigurationsdatei für den *mapwoc-redirector* Befehl.

Hier ein Beispiel für so eine Konfigurationsdatei:

```
[config]
ctrl_interface=localhost
ctrl_port=8228
redirector_interface=localhost
redirector_port=8123
#sslconfig=/var/lib/mapwoc/redirector/credentials
```

Alle Werte aus der Konfigurationsdatei lassen sich auch als Option über den Befehl *mapwoc-redirector* setzen. Hier ein Überblick aller verfügbarer **Optionen** (Auszug aus der Manpage von *mapwoc-redirector*, siehe auch `mapwoc-redirector --help`):

```
-v, --verbose
    Be verbose and chatty on stdout and logfile.

-q, --quiet
    Be quiet, only log warnings and errors.

--urllist
    Path to file containing URLs.

-p, --redirector_port
    Port on which to listen for http requests

--redirector_interface
    Interface on which to listen for http requests.

--ctrl_interface
```

```

    Interface on which to listen for commands.

-c,--ctrl_port
    Port on which to listen for commands.

--sslconfig
    Path to credentials for secured communication.

--config
    Path to configuration file to read configuration from
    (or write to if called with --generate-config).

--generate-config
    Generate a configuration file.

-h,--help
    Show help and exit.

--version
    Show version and exit.

```

5.4 Honey-Clients

Honey-Client-Systeme (HCS) werden über hcs-Konfigurationsdateien der jeweiligen mapWOC-Node bekannt gemacht.

Das `hcsdir` – das Verzeichnis mit den Konfigurationsdateien – wird in der `mapwoc-node.config` festgelegt (Voreinstellung: `/var/lib/mapwoc/node/hcs`).

Hier ein Beispiel für eine hc-Datei `winxp_ie6_demo.hc`:

```

[hc]
name=WindowsXP Demo Image IE6
description=Lots of other software on it
hcsid=winxp_ie6_demo
max_parallel=-1
controllable=true
active_browser=cpe://a:microsoft:ie6
installed_software=cpe:o:microsoft:etc

[control]
type=kvm
boots=true
image_master=/usr/share/mapwoc/winxp-master.img
image_file=/usr/share/mapwoc/images/winxp_ie6_demo.ovl
image_memory=128

[network]
own_ip=10.0.0.4
dns_ip=10.0.0.3
proxy_ip=10.0.0.2
gateway_ip=10.0.0.1
homepage=http://10.0.0.1/winxp_ie6_demo

```

Hier ein Überblick der verfügbaren Parameter aus der Konfigurationsdatei eines Honey-Clients (HCs):

```

[hc]
name=
    Name des HCs.
description=
    Beschreibung des HCs.

```

hcsid=
ID des HCs. Muss mit dem Dateinamen der HC-Konfigurationsdatei
übereinstimmen.

max_parallel=
Gibt an, wieviele Instanzen von diesem HC parallel gestartet
werden können. Der Wert "-1" (= unendlich) wird für virtuelle
HCs verwendet, "1" für native Geräte.

controllable=
Gibt an, ob der HC von mapWOC steuerbar ist. Bei virtuellen
HCs wird "true" verwendet, bei nativen HCs "false".

active_browser=
Angabe des aktiven Browsers (in CPE-Form).

installed_software=
Angabe der installierten Software (in CPE-Form). Mehrere
Einträge durch ";" trennen.

[control]
type=
Angabe des HC-Steuerungstyps. Mögliche Werte:
- 'kvm' (KVM-Image)
- 'kvm-mac' (KVM-Image mit MacOSX, erfordert spezielle
KVM-Startparameter)
- 'native' (nativer HC, wird manuell gesteuert)

boots=
Diese Variable muss "true" sein, wenn sich das KVM-Image im
Ruhezustand ("Suspend-to-disk") befindet. Andernfalls kann der
'boots'-Parameter weggelassen werden.

image_master=
Angabe des KVM-Master-Images (absoluter Pfad).

image_file=
Angabe des KVM-Overlay-Images (absoluter Pfad).

image_snapshot=
Angabe der KVM-Live-Migration-Snapshot-Datei (absoluter Pfad),
sofern vorhanden.

image_memory=
Angabe des durch den HC benötigten Arbeitsspeicher (in MB).

kvm_kernel=
Angabe eines speziellen KVM-Kernel Parameters zum Starten. Für
KVM-MacOSX-Images wird z.B. benötigt:
"kvm_kernel=/root/kvm-x/boot-osx-r327". (Details siehe manpage
von KVM, Option '-kernel').

interface=
Bei Nutzung eines nativen HCs muss hier das VLAN-Interface
angegeben werden, z.B.: 'vlan101'.

[network]
own_ip=
IP-Adresse des HCs.

dns_ip=
DNS-Adresse des HCs

proxy_ip=
Proxy-Adresse des HCs

gateway_ip=
Gateway-Adresse des HCs

homepage=
Startseite des Browsers (Redirector mit hcsid), z.B.:
'http://10.0.0.1/winxp_ie6_demo'

client_mac=
MAC-Adressen des nativen HCs

wlan_ssid=
SSID des zu verbindenden WLAN-Netzes für einen nativen HC.

[keyboard-shortcuts]

homepage=

Optionaler Tastatur-Kurzbehl, um die Browser-Startseite aufzurufen. Voreingestellt ist bereits 'alt-home' (=Alt+Pos1). Anpassungen nur für bestimmte Browser/Betriebssysteme nötig. Z.B. für Safari unter MacOSX: '0xDB-shift-h' (=CMD+Shift+h)

5.5 Weitere relevante mapWOC-Dateien

Die nachfolgend genannten Pfade zu den Dateien sind Voreinstellungen von mapWOC. Änderungen der Pfade und Dateinamen sind über die zugehörigen Konfigurationsdateien von Master bzw. Node möglich. Die Angaben *[Master]* und *[Node]* sollen bei der Unterscheidung helfen, ob es sich um eine Datei des Masters oder der Node handelt.

- *[Master]*: Die **SQLite-Datenbank** liegt unter

`/var/lib/mapwoc/db`

Die Datenbank existiert nur auf dem Master.

- *[Node]*: Das **Arbeitsverzeichnis der Node** liegt unter:

`/var/lib/mapwoc/node/work/`

Hier werden die TCP-Dumps (im pcap-Format) von jedem KVM- und Proxy-Prozess für die Analyse abgelegt.

- *[Node]*: Das **Analysescript** liegt auf jeder Node unter:

`/usr/bin/mapwocanalyse.pl`

Dieses Skript analysiert Netzwerkverkehr aus den vorliegenden Dump-Dateien. Dabei werden gesuchte Dateien extrahiert (sofern vorhanden) und durch einen konfigurierten Virenschanner untersucht.

- *[Node]*: Die **Anti-Virenschanner-Konfiguration** werden in folgender Datei vorgenommen:

`/etc/mapwoc/antivir.cfg`

- *[Node]*: **Analysierte Ergebnisse** eines jeden URL-Besuchs werden von der Node hier als tar.gz-Archiv abgelegt:

`/var/lib/mapwoc/node/results/`

- *[Master]*: Der Master holt sich die bereitgestellten Ergebnisse von allen Nodes ab und legt sie alle gemeinsam in diesem **Master-Ergebnis-Verzeichnis** ab:

`/var/lib/mapwoc/master/`

CA für mapWOC aufsetzen

6.1 Hintergrund

Unkonfiguriert kommunizieren die mapWOC-Komponenten (z.B. *mapwoc-master* und *mapwoc-node*) **ohne Authentifizierung** und **unverschlüsselt**.

Alle Komponenten unterstützen jedoch TLS-Authentifizierung und Verschlüsselung.

Für die Einrichtung einer CA wird in der Regel ein neues Paket benötigt, unter Debian Squeeze heißt dieses *gnutls-bin*.

Eine lesbare, generelle Einführung in das Thema asynchrone Verschlüsselung bietet z.B. das [Gpg4win Kompendium](#).

6.2 Eine CA erstellen

Bemerkung: Es kann natürlich auch eine bereits bestehende CA oder Zertifikate benutzt werden, siehe dazu weiter unten "Einbindung der Zertifikate".

Bemerkung: Folgende Schritte sind in einem temporären und geschütztem Verzeichnis durchzuführen. Idealerweise findet dies nicht auf dem mapwoc-System sondern auf einem externen System statt.

Bemerkung: Es können keine Passwörter verwendet werden.

1. Schlüsselpaar für die CA erzeugen:

```
certtool --generate-privkey --outfile ca-key.pem
```

2. Zertifikat für die CA erzeugen:

```
certtool --generate-self-signed --load-privkey ca-key.pem --outfile ca-cert.pem
```

Bemerkung: Alle außer den folgenden Fragen können mit den Voreinstellungen beantwortet werden (Enter drücken):

The certificate will expire in (days): 1000

Does the certificate belong to an authority? (y/N): y

Will the certificate be used to sign other certificates? (y/N): y

Will the certificate be used for signing (required for TLS)? (y/N): y

Is the above information ok? (Y/N): Y

3. Schlüsselpaar für den mapwoc-redirector erzeugen:

```
export CRDP=/var/lib/mapwoc/redirector/credentials/  
certtool --generate-privkey --outfile $CRDP/key.pem
```

4. Zertifikat für den mapwoc-redirector erzeugen und signieren:

```
certtool --generate-certificate --load-privkey $CRDP/key.pem \  
--outfile $CRDP/cert.pem --load-ca-certificate ca-cert.pem \  
--load-ca-privkey ca-key.pem
```

Bemerkung: Alle außer den folgenden Fragen können mit den Voreinstellungen beantwortet werden (Enter drücken):

Common name: mapwoc-redirector

The certificate will expire in (days): 1000

Is the above information ok? (Y/N): Y

5. Schlüsselpaar für den mapwoc-master erzeugen:

```
export CRDP=/var/lib/mapwoc/master/credentials/  
certtool --generate-privkey --outfile $CRDP/key.pem
```

6. Zertifikat für den mapwoc-master erzeugen:

```
certtool --generate-certificate --load-privkey $CRDP/key.pem \  
--outfile $CRDP/cert.pem --load-ca-certificate ca-cert.pem \  
--load-ca-privkey ca-key.pem
```

Bemerkung: Alle außer den folgenden Fragen können mit den Voreinstellungen beantwortet werden (Enter drücken):

Common name: mapwoc-master

The certificate will expire in (days): 1000

Is the above information ok? (Y/N): Y

7. Schlüsselpaar für die mapwoc-node erzeugen:

```
export CRDP=/var/lib/mapwoc/node/credentials/  
certtool --generate-privkey --outfile $CRDP/key.pem
```

8. Zertifikat für den mapwoc-node erzeugen:

```
certtool --generate-certificate --load-privkey $CRDP/key.pem \  
--outfile $CRDP/cert.pem --load-ca-certificate ca-cert.pem \  
--load-ca-privkey ca-key.pem
```

Bemerkung: Alle außer den folgenden Fragen können mit den Voreinstellungen beantwortet werden (Enter drücken):

Common name: mapwoc-node

The certificate will expire in (days): 1000

Is the above information ok? (Y/N): Y

9. Schlüsselpaar für die gui erzeugen:

```
export CRDP=/var/lib/mapwoc/gui/credentials/
certtool --generate-privkey --outfile $CRDP/key.pem
```

10. Zertifikat für den mapwoc-node erzeugen:

```
certtool --generate-certificate --load-privkey $CRDP/key.pem \
--outfile $CRDP/cert.pem --load-ca-certificate ca-cert.pem \
--load-ca-privkey ca-key.pem
```

Bemerkung: Alle außer den folgenden Fragen können mit den Voreinstellungen beantwortet werden (Enter drücken):

Common name: mapwoc-gui

The certificate will expire in (days): 1000

Is the above information ok? (Y/N): Y

11. Sichern des CA-Schlüssels:

Der CA-Schlüssel ca-key.pem wird nun nur noch benötigt, wenn dem System neue Komponenten (z.B. eine neue Node) hinzugefügt werden sollen. Er ist von dem ausführenden System zu entfernen und separat gesichert aufzubewahren.

6.3 Einbindung der Zertifikate

Wichtig: Versichern Sie sich erneut, dass restriktive Rechte (nur Nutzer-lesbar) auf den u.g. Verzeichnissen und Dateien gesetzt sind.

1. CA-Zertifikat bekannt machen:

```
cp ca-cert.pem /var/lib/mapwoc/redirector/credentials/
cp ca-cert.pem /var/lib/mapwoc/master/credentials/
cp ca-cert.pem /var/lib/mapwoc/gui/credentials/
cp ca-cert.pem /var/lib/mapwoc/node/credentials/
```

1. Die Konfigurationsdateien der Komponenten anpassen:

In den Konfigurationsdateien ist folgende Zeile auszukommentieren:

```
#sslconfig=/var/lib/mapwoc/redirector/credentials
```

wird dann zu

```
sslconfig=/var/lib/mapwoc/redirector/credentials
```

Der Pfad (/var/lib/mapwoc/redirector/credentials) ist für die anderen Komponenten entsprechend anzupassen (für den *mapwoc-master* z.B.: /var/lib/mapwoc/master/credentials), bzw. in den Vorgaben bereits angepasst.

Bemerkung: In den credentials-Verzeichnis müssen sich drei nicht-leere Dateien befinden:

key.pem (Schlüssel der Komponente)

cert.pem (Zertifikat der Komponente)

ca-cert.pem (CA-Zertifikat)

Bemerkung: In der .ini Datei der GUI heißt die auszukommentierende Option nicht *sslconfig* sondern *master.sslconfig*, also z.B.

master.sslconfig = /var/lib/mapwoc/gui/credentials

Impressum

Copyright 2011 Intevation GmbH

Das mapWOC-Handbuch ist unter der [Creative Commons CC BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/) lizenziert.