To: CyberSpark Open Source Project
From: Sky
Date: 11/16/10
Subject: CyberSpark 9x12 migration to PHP

Red7 Communications, Inc.
PO Box 27591
San Francisco CA 94127-0591
USA
Phone +1 415.759.7337
web.red7.com

Overview— CyberSpark.net

At any given moment hundreds of writers, bloggers and journalists exercising their right to free speech are under attack online.

Attacks can force these sites offline and make it impossible for their stories to be told, make it incredibly expensive to stay online, and in many cases make it difficult or impossible to raise needed funds.

Critical steps are required to safeguard this precious right:

- Detecting attacks as early as possible,
- Getting things fixed, and/or
- Moving to a defensible position when attacks are imminent or ongoing.

CyberSpark provides services that detect such attacks and alert the appropriate parties...and those who can do something about it.

Open Sourcing CyberSpark

Deploying CyberSpark software and services more broadly is critical to fulfilling our mission, so beginning this month we're making our free and open source software easier to obtain and use. This means that in addition to using our free online services, webmasters and system administrators can install components from our software, modify them, improve them, and use them in the ways that best suit their own needs. And their improvements can be shared among members of a broader user community.

We also supply online services that supplement these open source components, and services for hire when large-scale projects are required. We can make connections to people who can help with almost any security-related need.

Cyberspark is a "first line of defense," monitoring sensitive sites and sending alerts when there are problems, then making connections to the people who can help.

Overview— open-sourcing CyberSpark.net

Cyberspark's monitoring software evolved from the Red7 *Knowledgebase* software of 2002, which was in turn conceptually based upon the KUIS knowledgebase software begun in 1998. The earlier software wasn't open source, but had been made available for repurposing and expansion in each case by both the organization and the programmer.[1]

The purpose of the original **KUIS** software was to spider and index a limited set of URLs on the web and send alerts based on changes in those monitored web pages. It essentially was a *bookmark* service much like the later digg.com.[2]

The **Red7** knowledgebase application of this software took on the additional roles of building a searchable database of web pages and sending alerts based on automated "agent-based" searches. It also added the ability to accept incoming email and add it to the database, and built a more human-friendly user interface. As a side-effect, it grew and benefited as the Red7 mixed-reality games system was constructed, based on this software, beginning in 2003.

Some time after 2005 the software was repurposed to address more focused monitoring tasks, and by early 2009 it had become the underlying layer of the "**CyberSpark.net"** free-speech and human rights monitoring software, incorporating some additional functions that can identify compromised ("hacked") pages.

Architecture

The new PHP-based system and the current Java-based system both run on Linux cloud servers. We have several *virtual* servers in use now, and additional servers are kept "in waiting" in case the primary servers fail or come under attack. The system can be scaled up or down and is relatively mobile in case of attack. A new server can be deployed in under an hour.

Alerts are sent by email and SMS (using email-to-SMS gateways provided by cellular carriers). The PHP-based system can send alerts through any SMTP system and we prefer highly-reliable systems such as gmail that accept encrypted connections.

Moving to PHP and Open Source

Java was a good language and platform for the development of the system, and provided an "industrial strength" environment in the days when large servers were the norm. But more and more programmers have moved toward PHP in recent years, and the simple write-test-write-test PHP architecture that's built right into a web server is more appealing for rapid deployment and continuing community development. So PHP is a better choice for open source, at least for this particular application.

Therefore, in the interest of making the monitoring-alerting system more accessible, we're going to rewrite the application in PHP and make it available as *open source* at no charge to the NGO, free-speech and security communities.

The migrated application will be PHP-only, and will not require any database or web server setup. It can run on a small, stripped-down and hardened Linux system.

The Migration

The monitors run on headless servers that do not accept incoming connections other than SSH (for control). There is no web server actually running on these servers. Execution of the PHP script is from *cron* or from a command line interface.

A basic PHP script will read the *properties* files that contain information on general system setup and the URLs to be monitored. As it encounters each URL specification, it interrogates the URL, notes any problems, sends alerts if required, and goes to the next URL. Everything is logged and some values are saved for later runs (in other words, there is some limited persistent memory related to each monitored URL).

Because each monitoring process is a separate PHP script, *cron* may run several such scripts at the same time, each isolated in its own address space and able to run concurrently.

Email notifications will go out through any SMTP server (using TLS or SSL for secure connections), which the implementer must provide. We currently use our own email server to send notifications, but will switch to gmail because or server might come under attack and we anticipate gmail being much more resilient.

Google Safe Browsing: We have already implemented a Google Safe Browsing *responder* that can accept requests for "good or bad" status on single

URLs and report back in real time. The code is open source Python code from *Google Code* and it conforms to the GSB 2.0 API specification.

Planned Additions

Logging and Performance Analysis: Each script records the results of its run in a flat log file. These files can be picked up, digested and the data inserted into a database for later retrieval and analysis. For the sake of simplicity, we won't run a database on the monitoring systems, which are designed to be lean-and-mean. Logs can be transferred and analyzed separately. We will write either:

- A program to read the flat logs and put them in a MySQL database; or
- A program to analyze flat logs and create various charts and tables.

Real-time Reporting: Once logs are going into a database, another program can be written to extract reports in "real time" and display them.

Google Safe Browsing: We could take the GSB responder mentioned above and make it available as a service to those who would implement their own monitoring-alerting systems.

A Specific Implementation Plan

A PHP script as being developed that can be run either from the command line or from a web page. In its basic mode it will conduct one full spidering operation, examining multiple URLs (pages or files), recording its results both in a log file and in a persistent "store." The script can also be run in a "threaded" mode where it executes its loop repeatedly until it is shut down...thus providing an unattanded or automatic mode of operation.

We will provide simple setup and documentation on how to use the system. Because it requires installation of a couple of supporting PHP packages, it is intended for use by system administrators who have access to command-line tools, and probably won't be installable by inexperienced webmasters.

Notes, References, Credits

[1] KUIS: Knowledge Universe Interactive Studio. Joe Miller, Founder and President of KUIS, which was owned by Knowledge Universe, verbally granted the right to use the Perl-based software in perpetuity, to Jim Schuyler, its designer and programmer. Red7 Communications, Inc. is a California corporation 80% owned by Jim Schuyler, which has developed a Java-based extension of the KUIS concepts and used it as a business tool, offered it to nonprofit NGOs, and operated it as a private knowledgebase tool. Red7 operates CyberSpark.net as a *public benefit project*, and is open sourcing the new PHP version of the monitoring software under a Creative Commons by-nc-sa license.

[2] digg.com has morphed significantly since its origin, but was originally a bookmark-saving and bookmark-sharing online service. Bookmark preservation and organization was the fundemantal purpose of the original KUIS knowledgebase.