Platform-independent and extensible IDS with Prelude

Alert Center

The Prelude security information management system receives both host- and network-based IDS messages and displays them in an easy web interface. We show you how to set it up.

By David Rupprechter

The ability to react at the right moment can sometimes save lives. Intrusion Detection and Prevention Systems (IDS/IPS) help administrators in state-of-the-art datacenters evaluate logfields and messages from many different devices and operating systems, identify intruders quickly, and initiate the right action.

Hybrid System: Prelude

The distinction between host-based IDS (HIDS) and network-based IDS (NIDS) is important. A host-based IDS, such as Tripwire, typically only evaluates files and detects local attacks, whereas a network-based IDS, like Snort, can also search network traffic for signs of illicit activity.

Prelude [1] is a hybrid IDS that combines the benefits of both HIDS and NIDS and impresses with its feature richness, including a neat web-based administration interface and numerous client agents. The developers have pushed Prelude development in the past few years, creating a mature product that is capable of handling large-scale environments. Because of its plugin architecture and features such as the Correlator Engine, experienced administrators can also create their own flexible add-ons to facilitate their daily work.

Libprelude API

The Prelude library also lets users integrate existing programs with Prelude. Libprelude provides APIs for various programming languages [2]. Making this integration

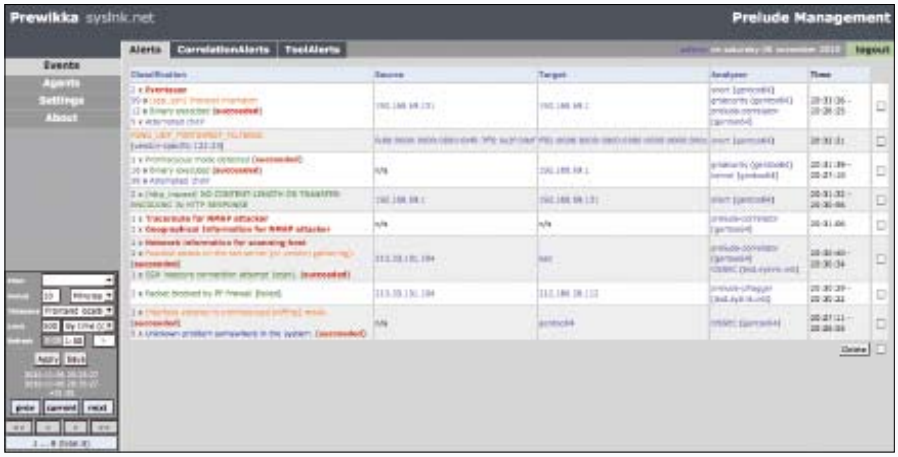
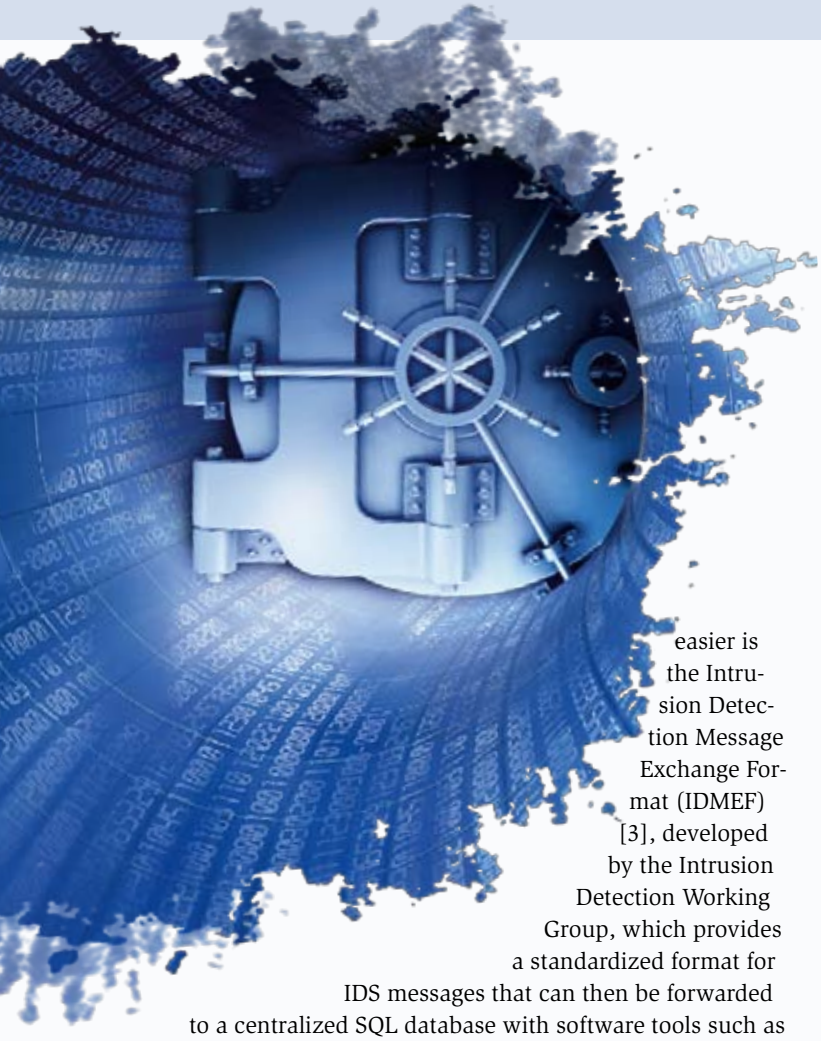


Figure 1: The Prelude web interface goes by the name of Prewikka. It gives users a tabular overview of all the events reported by the various agents.

THE AUTHOR

David Rupprechter is an IT technician with a leading Austrian system integrator and focuses on IT security, network technology programming, and Linux at *http://www.dotlike.net*.



easier is the Intrusion Detection Message Exchange Format (IDMEF) [3], developed by the Intrusion Detection Working Group, which provides a standardized format for IDS messages that can then be forwarded to a centralized SQL database with software tools such as those supported by the Libprelude API. For administrators, this approach removes the need to correlate log entries from the network because the Prelude agents send their data to a centralized manager via encrypted SSL connections. The manager adds the data to the database in real time and provisions the data for querying. Heartbeats sent to the agents tell the manager whether the connection is up. In the case of an interruption, the agents cache any entries they have not transmitted then send the missing results when the connection is resumed.

Agents are available for the following platforms and others:

- Network-based Snort IDS [4]
- Host-based Open Source Security (OSSEC) [5]
- Linux system logs via Prelude-LML [6]
- PAM authentication

The Security Information Management (SIM) system also can analyze logs from typical hardware appliances and services, including various Cisco appliances, Exim, Microsoft SQL, ClamAV, Nessus, Apache, Tripwire, Netfilter, ipchains, and ipfw.

OSSEC adds rootkit detection, Windows Registry monitoring, and file integrity checking. All of these features are available independently of the platform for Linux, Windows, VMware ESX, BSD, Solaris, AIX, HP-UX, and Mac OS X. Additionally, remote syslog integrates hardware appliances such as the Cisco PIX/ASA, Juniper NetScreen, and SonicWALL firewalls.

Prelude Manager

As the central authority for the four Prelude components (Manager, sensors, Libprelude, and web interface), the Manager accepts the data that the sensors transfer from the connections managed by the Libprelude library. The Prelude Manager is easily installed on Debian; typing `aptitude install prelude-manager libprelude` takes you to a dpkg configuration dialog, where you can select the database (MySQL or PostgreSQL) and enter the SQL root password. The system then generates the SSL key.

After setting up the database and creating a manager profile, you can modify the configuration in the `/etc/prelude-manager/prelude-manager.conf` file. You will need to modify the `listen` entries to match your own needs and replace the loopback address of 127.0.0.1 with your own IP to allow the computer to receive logs from remote agents.

Also, you will need to modify the SQL configuration in the `[db]` section:

```
[db]
type = mysql
host = localhost
port = 3306
name = Datenbankname
user = Benutzername
pass = Passwort
```

On Debian GNU Linux, you will need to set the `RUN` variable in `/etc/default/prelude-manager` to `YES`, which launches the Prelude Manager with the `/etc/init.d/prelude-manager start` command.

Prewikka

Prewikka, the Prelude web interface, is useful for displaying the information provided by what can be a large number of agents (Figure 1) [7].

To install Prewikka on Debian, type `aptitude install prewikka` at the command line and complete the configuration tasks. For example, you'll need a database, which you can install with the use of the `/usr/share/prewikka/database/mysql.sql` script.

This step will put your Prewikka configuration file in `/etc/prewikka/prewikka.conf`. Then, you'll need to set the `software`, `place`, and `title` variables to suit your own requirements – for example:

```
software: Prewikka
place: dotlike.net
title: Preludeconsole
```

PREWIKKA AS A VIRTUAL HOST

If you prefer to run Prewikka's web interface on Apache, you can add the lines shown in Listing 1 to your `/etc/apache2/site-available/` directory on Debian.

To enable the host, you'll need the `a2ensite Vhost-file` command. Putting this into a separate file and using an `Include` directive in `00_default_vhost.conf` is also a useful option.

Next, type `/etc/init.d/apache restart` to restart the web server. Of course, the Prelude server needs a name record that DNS can resolve. Depending on your network infrastructure, you also might need an A Record for `preludehost` on your DNS server for this.

The following section in the configuration file controls access to Prewikka:

```
[database]
type: mysql
host: localhost
user: prewikka
pass: Passwort
name: prewikka
```

The Debian package configuration typically gives you useful defaults. You can now launch `prewikka-httpd` to start the web interface with its built-in web server on port 8000. Administrators who prefer to use an existing Apache installation should check out the "Prewikka as a Virtual Host" box for a short how-to.

The default username and password for Prewikka are both `admin`. You should change this in *Settings* | *My Account* when you first log in. The web interface is intuitive and gives you a clear overview of alerts. Filter options such as source, target, analyzer, or classification are available for all fields. The menu in the bottom right lets you set the period and the number of events per page.

Prelude Sensors: LML

One of the most important Prelude sensors for Linux administrators is the Prelude Log Monitoring Lackey (LML) [6]. It runs as a host-based IDS and monitors defined logfiles, sending events to the Prelude Manager. To install LML on Debian, type `aptitude install prelude-lml`.

After installing any sensor, you must register it with the Prelude Manager via an encrypted connection. For Prelude-LML, do the following:

```
prelude-admin register prelude-lml U "idmef:w admin:r" ➥
   List_of_IP_Prelude-Managers -uid 0 -gid 0
```

After a short wait, Prelude prompts you to run the `prelude-admin registration-server prelude-manager` command in a second shell. This step provides a one-shot password that the sensor uses to authenticate against the existing console session. Next, you'll see the following command at the console running the command, and you can confirm by pressing `y` or Enter.

```
Approve registration? [y/n]: y
```

Changing the server address in the `/etc/prelude/default/client.conf` configuration file (`server-addr = IP` line) ensures that the local agents will send their data to the specified server in the future.

LISTING 1: Prewikka Virtual Host Configuration

```
01 <VirtualHost 192.168.0.1:80>
02 ServerName preludehost
03 ErrorLog /var/log/apache2/error_log
04 CustomLog /var/log/apache2/access_logcombined
05   <Location "/">
06    AllowOverride None
07    Options ExecCGI
08    <IfModule mod_mime.c>
09      AddHandler cgi-script .cgi
10    </IfModule>
11    Order allow,deny
12    Allow from all
13   </Location>
14 Alias /prewikka/ /usr/share/prewikka/htdocs/
15 ScriptAlias / /usr/share/prewikka/cgi-bin/prewikka.cgi
16 </VirtualHost>
```

Snort

To integrate the popular Snort software with Prelude, you'll need to enable signature-based network intrusion detection. Because the developers are continually expanding Snort's ruleset, administrators must keep the intrusion signatures up to date. Following free registration on the Snort website [4], you will receive a code for automated signature updates that become available 30 days after official publication. Third-party programs like Oinkmaster [8] or Pulled Pork give you the ability to automate the signature update downloads using a cronjob.

Bleeding Snort [9] is an alternative to the official Snort rules, and something similar to this is used by commercial Sourcefire appliances [10]. The project website continually publishes Snort rules created by network and security specialists. Additionally, you are free to combine the two rulesets.

To install Snort with a MySQL database in the background, type `aptitude install snort-mysql`, choose `Systemstart` as the start method, and disable `Promiscuous Mode`. In this mode, Snort checks every single package, even if it is not directly targeted at the IP address of the network interface. Email logging is optional, and your choice of log database will now obviously be Prelude.

Sniffer Configuration

Now you need to apply a number of changes to the `/etc/snort/snort.conf` configuration file. To tell Snort to send its data to Prelude, add the following section:

```
output alert_prelude: profile=snort
```

The following settings define the network to monitor:

```
# for IPv4:
var HOME_NET 192.168.0.0/24
# for IPv6 USE flag:
ipvar HOME_NET 192.168.69.0/24
# The external network can stay set to any
var EXTERNAL_NET any
```

Snort's functionality – such as the ability to detect portscans – depends on a collection of preprocessors. You can optimize them with granular settings, and it thus makes sense to go through the whole of the configuration file. Snort also needs a Prelude profile:

```
prelude-admin register snort "idmef:w admin:r" ➥
   List_of_IP_Prelude-Managers -uid 0 -gid 0
```

The profile shown, for example, will allow Snort and Prelude to cooperate.

OSSEC

If platform independence is important to you, the host-based OSSEC IDS is an obvious choice. Once you've obtained the latest version [5], install the Prelude developer libraries you require for the build with `aptitude install libprelude-dev`. After unpacking the tarball in the `src` subdirectory, type `make setprelude` to enable Prelude support and then `install.sh` to start the install.

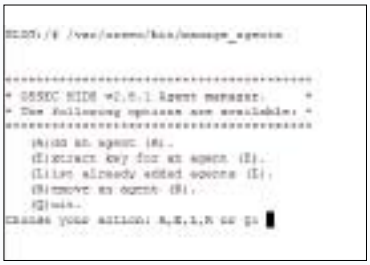


Figure 2: The "manage_agents" script provides a simple menu that administrators can use to register client agents on the server.

A local installation will be fine for users who do not plan to integrate hardware terminal devices or other clients, but typically you will want to select a `server` and enable the Integrity Check Daemon and the Rootkit Detection Engine. To provide protection against brute force attacks, enable Active Response and Firewall Drop to block the attacking hosts automatically via iptables, ipchains, or ipfw, depending on the system in question. To avoid issues with false positives on the network, you might also want to consider an IP address whitelist. The next step is to enable the receipt of syslog messages if you plan to use syslog to integrate hardware appliances.

Now you can modify the `/var/ossec/etc/ossec.conf` file to tell OSSEC to hand over the information to Prelude:

```
<ossec_config>
<global>
  <email_notification>no</email_notification>
  <prelude_output>yes</prelude_output>
</global>
```

Finally, you need to register OSSEC with Prelude, too:

```
prelude-admin register OSSEC "idmef:w admin:r" ➥
   List_of_IP_Prelude-Managers -uid ossec -gid ossec
```

Because the analysis service runs under this user account, you need to use `ossec` as the UID and GID here. After completing the install, you can launch the HIDS by typing `/var/ossec/bin/ossec-control start`. Before an agent can connect with OSSEC, you need to register it on the server. The `/var/ossec/bin/manage_agents` command lets you do so in a keyboard-controlled console menu (Figure 2).

After entering the name, you need to enter the IP address. If the address has been assigned dynamically, just specify the Class C network on which the agent resides (e.g., 10.21.81.0/24). Then, select an ID for the agent; the default suggestion for the first agent will be `001`, and there's no need to change it. To bind the agent to the hardware device, you also need an authentication key. To create one, type `/var/ossec/bin/manage_agents -e 001`.

Windows and Syslog

To integrate a Windows client with OSSEC and Prelude, you'll need Libprelude [11] on the system. Installation consists of a couple of clicks; you will probably want to select all the software components apart from *Source Code* (Figure 3).

After completing the install, the Windows administrator needs to run *Manage Agent* (in the start menu or the *OSSEC* folder). After entering the *OSSEC Server IP* and the matching authentication key, the Windows computer can open a connection to the host-based IDS (Figure 4). Clicking *Save* saves the data you entered; to fine-tune the configuration, go to *View* | *Config*. After rebooting, OSSEC will use a Windows service to open the connection.

OSSEC's syslog functionality is useful for hardware appliances like Cisco's PIX firewalls. This functionality is enabled by the installation process described previously, but the server will not receive any syslog messages by default. For this to happen, you must add to the configuration file `/var/ossec/etc/ossec.conf` the IP addresses from which message reception is allowed:

```
<remote>
  <connection>syslog</connection>
  <allowed-ips>10.21.81.4</allowed-ips>
  <allowed-ips>10.21.81.9</allowed-ips>
  <port>5140</port>
</remote>
```

Figure 3: When installing Windows agents for OSSEC, you need to select everything except *Source Code*.

LISTING 2: setup.py

```
01 From setuptools import setup, find_packages
02 setup(
03   name='PortscanGeoinfoPlugin',
04   version='1.0',
05   description="Geographical information for portscanning
                 hosts",
06   author="David Rupprechter",
07   packages=find_packages(),
08   entry_points='''[PreludeCorrelator.plugins]
       PortscanGeoinfoPlugin =
       portscangeoinfoplugin:PortscanGeoinfoPlugin'''
09 )
```

In this configuration, OSSEC processes syslog messages from 10.21.81.4 and 10.21.81.9.

The Prelude Correlator Engine [12] is really useful if you are trying to track down attackers because it gives you the option of correlating IDMEF events and thus generating correlation alerts that contain all the individual alerts involved (Figure 5). To install the Prelude Correlator on Debian, type `aptitude install prelude-correlator`, then you can register the agent as follows:

```
prelude-admin register prelude-correlator "idmef:rw" ➥
   List_of_IP_Prelude-Managers -uid 0 -gid 0
```

The Prelude Correlator comes with a number of plugins of its own.

On Debian systems, you will find these plugins in the `/usr/share/pyshared/PreludeCorrelator/plugins/` folder, and you can enable them by adding an entry in the `/etc/prelude-correlator/prelude-correlator.conf` file. For example, this entry:

LISTING 3: main.py

```
# Please download http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz and place the content of the archive in /usr/share/GeoIP/ to get more detailed geographical information

# Please report bugs to rupprechter at dotlike dot net
# for more information please visit www.dotlike.net/portscangeoinfo.php

import os
import re
import shlex, subprocess
import sys

from PreludeCorrelator.context import Context
from PreludeCorrelator.pluginmanager import Plugin
from PreludeCorrelator.idmef import IDMEF

strout = None

class PortscanGeoinfoPlugin(Plugin):
  def run(self, idmef):
    classification = idmef.Get("alert.classification.text")

    NMAPFSCAN = "PSNG_TCP_FILTERED_PORTSCAN"
    NMAPSCANUDP = "PSNG_UDP_FILTERED_PORTSCAN"
    NMAPPSNG = "ICMP PING NMAP"
    TCPPSCAN = "(portscan) TCP Portscan"
    NMAPSCAN = "PSNG_TCP_PORTSCAN"
    SSHSCAN = "SSH insecure connection attempt (scan)."
    SSHSCAN2 = "Possible attack on the ssh server (or version gathering)."
    if classification != NMAPPSNG and classification != NMAPFSCAN and classification != TCPPSCAN and classification != NMAPSCAN and classification != SSHSCAN and classification != SSHSCAN2 and classification != NMAPSCANUDP:
      return
    source = idmef.Get("alert.source(*).node.address(*).address")
    for saddr in source:
      ctx = Context(("NETWORK_INFO_PORTSCAN_ATTACKER", saddr), { "expire": 600, "alert_on_expire": True }, update = True, idmef=idmef)
      if ctx.getUpdateCount() == 0:
        ctx.Set("alert.correlation_alert.name", "Network scan of host(s) detected" )
        ctx.Set("alert.classification.text", "Network information for scanning host")
        ctx.Set("alert.assessment.impact.severity", "high")
      # only create one alert per source
      # GeoIP
      # only do GeoIP lookup if IP address is not private
        if saddr.find("10.", 0, 4) != -1 or saddr.find("172.", 0, 4) != -1 or saddr.find("192.", 0, 4) == -1:
          # if GeoIP City Database exists
          if os.path.exists("/usr/Share/GeoIP/GeoLiteCity.dat") == True:
            import GeoIP
            gi = GeoIP.open("/usr/share/GeoIP/GeoLiteCity.dat",GeoIP.GEOIP_STANDARD)
            geoinfo = gi.record_by_addr(saddr)
            geo = re.sub(r'\'', r'', re.sub(r': ', r':', str(geoinfo)) )
            idmef = IDMEF()
            idmef.Set("alert.classification.text", "Geographical Information for portscanning host")
            idmef.Set("alert.correlation_alert.name", geo )
            idmef.Set("alert.assessment.impact.severity", "high")
            idmef.Set("alert.assessment.impact.description", "Geographical information for portscanning host " + saddr)
            idmef.alert()
            ctx.update(idmef=idmef)
          # else if only GeoIP Country Database exists
          elif os.path.exists("/usr/share/GeoIP/GeoIP.dat") == True:
            import GeoIP
            gi = GeoIP.new(GeoIP.GEOIP_MEMORY_CACHE)
            idmef = IDMEF()
            idmef.Set("alert.classification.text", "Geographical Information for portscanning host")
            idmef.Set("alert.correlation_alert.name",str(gi.country_code_by_addr(saddr)) )
            idmef.Set("alert.assessment.impact.severity", "high")
            idmef.Set("alert.assessment.impact.description", "Geographical information for portscanning host " + saddr)
            idmef.alert()
            ctx.update(idmef=idmef)
```

1 x **Network information for scanning host** 1 x Possible attack on the ssh server (or version gathering). (**succeeded**) 1 x SSH Insecure connection attempt (scan). (**succeeded**)	213.33.101.194	bsd	prelude-correlator (gentoo64) OSSEC (bsd.sysink.net)

Figure 5: Correlation alerts will record attacks even if hackers target multiple devices; this example shows a portscan.

```
[BruteForcePlugin]
disable = false
```

enables the brute force plugin. To start the Correlator, you can then type `/etc/init.d/prelude-correlator start`.

Your Own Plugins

If you are interested in creating your own plugins, you will need some Python skills, and you should also read Chapter 4.2 ("The Message Classes") of the IDMEF-RFC [3]. This section of the RFC relates to the message classes that plugin programmers are allowed to use.

The PortscanGeoinfo plugin, which I wrote as an example, is available from dotlike.net [13]. This program reacts to portscans that have been detected by Snort and OSSEC and lets the administrator output the geographical information for the attacker via a correlation alarm. To allow Snort to detect portscans, the portscan preprocessor must be enabled in your Snort configuration file, `snort.conf`.

The *Network Information for scanning host* correlation alert contains both the geographical IP information and the matching Snort and OSSEC sensor alerts. The plugin waits for five minutes, collects all the portscan events within this period of time, then bundles them into a correlation alert. To allow all of this to happen, you need to install GeoIP for Python up front by typing `aptitude install geoip-python libgeoip1`. If you are not happy with the country information, the GeoIP City Lite database [14] will give you more detail.

Plugin Structure

Each correlation plugin contains a file called `setup.py` (Listing 2), which is responsible for installing the plugin. To create your own plugin, just modify the `PortscanGeoinfoPlugin` and `portscangeoinfoplugin` names and optionally correct the `description`, `author`, and `version` fields.

In my example, the plugin code is stored in the `portscangeoinfoplugin` folder. The script will also accept wildcards here. Inserting the following:

Figure 4: In contrast to syslog integration, the Windows agent uses a secure encrypted connection to transfer its data by default.

```
Pluginname = *
  Directory:My_plugin
```

into the setup script will reference the `My_plugin` plugin in the `Directory` folder. Then `python setup.py build` and `python setup.py install` will complete the install. You now need to add the following lines to the file `/etc/prelude-correlator/prelude-correlator.conf` to enable the plugin when the Correlator is launched:

```
[PortscanGeoinfoPlugin]
disable = False
```

The plugin folder contains the `init.py` and `main.py` files (see Listing 3), and the `init.py` file is a one-liner with the plugin name:

```
from main import PortscanGeoinfoPlugin
```

The PortscanGeoinfo plugin shows you the amazing options that the Prelude Correlator Engine offers: The Popen function integrates command-line programs and their output.

Full Control of the Network

With the combination of NIDS and HIDS as Prelude sensors, you can analyze all the information your network gives you – including information from syslog-capable terminal devices – in a centralized web interface.

Whether you have Windows, Cisco, or Linux servers, this setup allows you to integrate any machine in your datacenter with your centralized monitoring instance.

If you then use individual plugins for displaying bundled information via correlation alerts in Prewikka, you have all the tools you need to make it very difficult for any intruder to sneak onto your network. ■■■

INFO

[1] Prelude: *http://www.prelude-technologies.com/en/solutions/universal-sim/index.html*

[2] Libprelude: *https://dev.prelude-technologies.com/wiki/prelude/ManualDevel*

[3] RFC 4765: IDMEF: *http://www.ietf.org/rfc/rfc4765.txt*

[4] Snort: *http://www.snort.org*

[5] OSSEC: *http://www.ossec.net*

[6] Prelude-LML: *https://dev.prelude-technologies.com/wiki/1/PreludeLml*

[7] Prewikka: *https://dev.prelude-technologies.com/wiki/1/Prewikka*

[8] Oinkmaster: *http://oinkmaster.sourceforge.net*

[9] Bleeding Snort: *http://www.bleedingsnort.com*

[10] Sourcefire: *http://www.sourcefire.com*

[11] Libprelude Windows agent: *http://www.prelude-technologies.com/download/releases/libprelude/libprelude-1.0.0.exe*

[12] Correlation Engine: *http://www.prelude-technologies.com/en/solutions/correlation-engine/index.html*

[13] PortscanGeoinfo: *http://www.dotlike.net/portscangeoinfo.php*

[14] GeoIP city database: *http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz*

[15] Code listings for this article: *http://www.linuxpromagazine.com/Resources/Article-Code*