# Puppet ชั้นที่ 8 ของลินุกซ์ - ฉบับที่ 165 สิงหาคม 2009

**โดย** Lisa Kachold **[1]**

แปลโดย Sake [2]

## Puppet - ระบบความปลอดภัยง่าย ๆ สำหรับผู้ใช้, นักพัฒนา, และผู้ดูแล

การดูแลรักษาการตรวจสอบด้านความปลอดภัยจำนวนมาก สามารถทำให้เกิดความกลัวได้บนพื้นฐานโดยทั่วไป.
ปัญหากับตัวตรวจสอบบันทึกเหตุการณ์ด้านความปลอดภัย แบบเสือใส่ในกระป๋อง และข้อมูลบันทึกเหตุการณ์รายวันอื่น
ที่ได้จากระบบไม่ได้จำเป็นที่จะเฉพาะเจาะจงถึงการใช้ระบบ. และ, เชื่อฉันเถอะ,
ฉันสงสัยว่า ผู้ใช้ใด ๆ, นักเขียนโค้ด, หรือผู้ดูแลระบบ ได้กำหนด "เวลาในการอ่านบันทึกเหตุการณ์" อย่างเพียงพอแล้ว
บนพื้นฐานว่า มีอะไร และ การเกิดคุกคาม. เพราะฉะนั้น, นโยบายที่ดีที่สุดคือ ปรับแต่งค่าคอนฟิกูเรชัน,
เว้นแต่การคอนฟิกมากกว่าหนึ่งระบบ อาจต้องใช้เวลาอันมีค่าไป. สำหรับกรณีนี้ เรามี Puppet และสูตรในการปรับแต่งค่าคอนฟิกูเรชัน.
เมื่อเรามีข้อมูลระบบโดยปกติ, ปรับแต่งค่าคอนฟิกูเรชันในแบบที่เราสามารถใช้งานมันได้,
เราสามารถใช้เวลาเป็นสัปดาห์ หรืออย่างนั้นการปรับแต่งค่าคอนฟิกูเรชันของตัวกรองอีเมล์ด้วยสคริป bash/cron หรือ ตัวกรอง Google
ที่จะ "แจ้งเตือนเราจริง ๆ" เมื่อสิ่งที่น่ากลัวได้เกิดขึ้น.

การแลกเปลี่ยนเล็กน้อยได้ถูกสร้างในการการดูแล, พัฒนา, และการใช้, ในการนั้น เรา
ปรับค่าคอนฟิกูเรชัน "เพียงทำให้น้อยที่สุด"เมื่อเราได้กำลังแข่งอยู่ตลอดกับเวลา สำหรับบริการใด ๆ ก็ตามที่ระบบลินุกซ์
ได้ถูกบีบรวมเข้าด้วยกัน. นี่อาจจะเป็น Twitter, YouTube, และ GMail,
Eclipse/Maven, หรือระบบเครื่องแม่ข่ายสำรองข้อมูล.
สิ่งเหล่านี้เป็นคำตอบแบบ s-hexy ที่จะอนุญาตให้ ผู้ดูแลระบบลินุกซ์ ดูแลมากกว่า 100 ระบบการการผลิต(GoDaddy.com, Dotster.com, Google.com)
ในแบบที่ยังปลอดภัย,ประหยัดค่าใช้จ่าย,และมีประโยชน์.

ผู้ดูแลระบบก่อนหน้านี้ สร้างเครื่องแม่ข่ายทั้งหมดโดยที่ไม่มี NTP (และใช้เวลา UTC ที่ต่างกันในการเปรียบเทียบกับเวลาระบบ) รีเปล่า?
Puppet สามารถดึงสายตัวอักษรนี้ได้.

ยกตัวอย่างเช่น กับไฟล์ /etc/sudoers ของคุณ, ในระบบความปลอดภัยลินุกซ์ชั้นที่ 8,
การทำ "อย่างน้อยที่สุด" แบบง่าย ๆ ไม่ได้ดีพอ! ต่างกับความสามารถของ Puppet.

## ส่วนประกอบทั้งหมดของ Puppet - ปลอดภัยในการใช้งานจริงหรือ?

เนื่องจากว่า Puppet จะทำการลงโดยมใช้ Ruby (และสามารถถูกขยายด้วย Ruby Gems),
มันไม่ได้รวมไบนารี SUID ใด ๆ มาด้วย. ดังนั้น, ผู้ดูแล Puppet ได้เพิ่มความเสี่ยงเล็กน้อย
ในสภาวะการใช้งานจริงทั้งหมด. Puppet ใช้ ใบรับรอง OpenSSL. ดังนั้น, เราสามารถแน่ใจได้ว่า
ข้อมูลต่าง ๆ มีความปลอดภัยตามการเข้ารหัสปัจจุบัน.

## คำตอบการปรับแต่งค่าคอนฟิกูเรชันแบบรวดเร็ว

ตอนนี้, การมีความสามารถในการนำสูตรไปใช้จากเครื่องเครื่องแม่ข่ายเฉพาะเครื่องเดียว,
ปรับค่าคอนฟิกูเรชันทีเดียว และจัดเตรียม หรือใช้หลาย ๆ ครั้ง, เปลี่ยนรูปแบบที่ผู้ใช้, นักพัฒนา, และผู้ใช้ระบบอย่างจริงจัง.
สมมติว่า, CERT ได้ประกาศการการโจมตีไปยังบริการที่ถูกเปิดอยู่ในส่วนหนึ่งของระบบ DMZ ของคุณ,
ดังนั้น, แทนที่จะทำการสร้างรายการบรรทัดคำสั่งของ iptables แบบง่าย ๆ, คุณสามารถกำหนด
มาตรฐานการทำงานอย่างสมบูรณ์, ปรับค่าคอนฟิกูเรชันของตัว Shorewall 3.0 ได้อย่างเต็มที่.

/etc/ssh/sshd_config ได้ตั้งค่าอย่างถูกต้องหรือเปล่า? ทุก ๆ ผู้ดูแลระบบ รู้ว่าถึงปริมาณอย่างมากของงาน
ที่จำเป็นสำหรับการเปลี่ยนนโยบายง่าย ๆ, ตัวอย่างเช่น ปีก่อนนี้ ตามที่การบุกรุกโดยการใช้กุญแจบนพื้นฐานของ ssh มีการประกาศอย่างแพร่หลาย.
Puppet สามารถทำสิ่งนี้ระหว่างเครื่องแม่ข่ายในบ้านทั้งสองของคุณ ถ้าคุณต้องการที่จะสามารถ
เพ่งความสนใจในการพัฒนางานได้.

การจัดการกุญแจ เป็นงานที่ใหญ่ยิ่งสำหรับเครื่องแม่ข่ายหรือผู้ใช้ที่เกี่ยวข้องทั้งหมด เว้นแต่กับ Puppet.

Nagios เป็นเครื่องมือที่ดีเยี่ยมสำหรับการเฝ้าดูระบบ, แต่ทว่ามีการปรับค่าคอนฟิกูเรชันอันยาวเหยียดอย่างไม่น่าเชื่อ ตามเครื่อข่ายที่ใหญ่มาก ๆ,เว้นแต่กับ Puppet.

แทนที่จะใช้ freshclam ในการให้ปรับปรุง ClamAV ในระบบของคุณ,
ใช้ Puppet.

แล้วการจัดการรหัสผ่านล่ะ? เชื่อหรือไม่, ระบบที่ใช้งานจริงหลาย ๆ ระบบลบผู้ใช้ที่เข้าถึงโดย VPN แบบง่าย ๆ,
เมื่อความสัมพันธ์ในการทำงานได้ถูกให้บริการ.
ผู้ใช้ทั้งหมดจากการใช้งาน 5 ปีผ่านมา (เสียงครวญครางหรือยิ่งกว่านั้น)
ของประวัติของระบบทั้งหมดก็ยังอยู่ในไฟล์รหัสผ่าน? ดังนั้น
ท่อ SSH -L ขาออกจาดพอร์ต 80/443 จาก anacron/cron หรืองานพิเศษ ก็ยังทำงานอยู่อย่างสบายใจ.
Puppet สามารถจัดการ ผู้ใช้/รหัสผ่าน อย่างถูกต้องอย่างง่ายดาย.

ยังไม่ปรับแต่งคอนฟิกกูเรชันของที่เก็บ YUM อย่างถูกต้อง เนื่องจากความยากของงานในการเข้าไป หรือ
ระเบิดการปรับแต่ง และการทดสอบกว่า 30 เครื่องแม่ข่ายรีเปล่า? การสำรองข้อมูลเป็นสิ่งเล็กน้อยด้วย Puppet.

รายการเครื่องแม่ข่าย DNS ของคุณได้ถูกเปลี่ยนโดยการค้นการค้นพบจากการวนซ้ำ หรือ
การโจมตีแบบพลีชีพอื่นใดไปยัง BIND รีเปล่า? ไม่มีปัญหากับ Puppet; คุณสามารถเปลี่ยนสิ่งเหล่านี้ได้อย่างรวดเร็ว.

คุณใช้งาน Linux/Solaris รีเปล่า? Puppet จะทำการปรับแต่งคอนฟิกูเรชันเข้าระบบโดย CDE (Common Desktop Environment).

และ Puppet สามารถที่จะถูกปรับแต่งคอนฟิกูเรชันไปยังการตั้งค่าหลากหลายด้วยตัวมันเอง ไปยังระบบใหม่, เมื่อคุณได้ตั้งค่าการเชื่อมต่อของคุณ.

คุณต้องการที่จะเปลี่ยนรายการ /etc/motd เพื่อที่จะเพิ่มป้ายความปลอดภัยไปยังกว่า
100 เครื่องหรือเปล่า? คุณต้องการที่จะเปลี่ยนที่อยู่อีเมล์บน index.html หรือ ปรับปรุงไฟล์ .htaccess ใหม่บนกลุ่มเครื่องแม่ข่ายรีเปล่า?
คุณต้องการที่จะเปลี่ยนสคริป cron รีเปล่า? Puppet สามารถที่จะแก้ไฟล์ข้อความได้เช่นเดียวกัน.

จินตนาการถึงความเป็นไปได้สิ.

### การติดตั้งแบบรวดเร็ว

การติดตั้งประกอบไปด้วย Facter [3] กัย Ruby, และที่จะใช้ Ruby Gems [4] ก็ได้.

แนะนำให้อ่าน : Install Puppet [5]

### สูตร

OpenNTPD [6]
File Permission Check [7]
Sudo [8]
Centralized Sudoers [9]

Apt Keys [10]
Module Iptables [11]
Shorewall 3.0 [12]
SSHD Config [13]
Nagios [14]
Authorized Keys [15]
ClamAV [16]
User Home Recipes [17]
Password Management [18]
Firmware Password [19]
Yum Server Build [20]
ResolvConf DNS [21]
Solaris CDE Login [22]
Zabbix Agent [23]
Puppet Install [24]
Simple Text [25]

Puppet ยังใหม่อยู่, แต่แนวคิดรวบยอดไม่ใช่; cfengine และเครื่องมืออื่น ๆ นั้นก็ยังมีอยู่.
อย่างไรก็ตาม, Puppet ได้ง่ายที่สุด และมีความสามารถสูงอย่างชัดเจน สำหรับที่จะใช้ในระบบ
สภาวะแวดล้อมแบบ *nix แบบสมบูรณ์. คาดหวังสิ่งที่ดีเยี่ยม, อย่างที่เครื่องมือนี้มีให้..

"??????????????????????????????????, ????????????????. ( Make everything as simple as possible, but not simpler.)"
 - ????????? ?????????

หมายเหตุ: ถ้ารูปแบบการจัดการระบบของคุณเป็นแบบ Reactive, มีต้นกำเนิดจาก Trenches Dot Com University หรือ คอร์ส Crisis Junkie 101, และ/หรือ คุณกำลังที่จะถูกไล่ล่า เพียงแค่ "ทำมันเดี๋ยวนี้, และอย่างรวดเร็ว"( ความไม่ปลอดภัย จึงเรียกว่าแนวทางแบบ "แรงผลักจากผลประโยชน์"), จุดสนใจทั้งหมดของคุณจะถูกเปลี่ยนโดยมูลฐาน. ที่ทำงานของคุณ จะมีเวลาอันมีความมากขึ้น ในการที่จะทำสิ่งที่ดีกว่า. เมื่อคุณกำลังจะดึงสายอักขระออกมาจากระบบของคุณ.

---

บทความนี้ ถูกให้บริการแบบสองความรับผิดชอบ เป็นส่วนประกอบของการนำเสนอสำหรับ
Phoenix Linux Users Group August HackFest: August 8, 2009 ที่ The
Foundation for Blind Children, 10 AM - 1 PM.

อ้างอิง PLUG : http://plug.phoenix.az.us/node/659 [26] หรือเพียงแต่ใช้
http://plug.obnosis.com/ [27].

แล้วเจอกันที่นี่!

Talkback:
พูดคุยเรื่องบทความนี้กับ The Answer Gang [28]

---

Lisa Kachold เป็นผู้ดูแลระบบ/ความปลอดภัยบนลินุกซ์, ผู้ดูแลเว็บ,
inactive CCNA, และผู้เขียนโปรแกรม และกว่า 20 ปีกับประสบการณ์ในการใช้งานจริงกับลินุกซ์.
Lisa ผ่านการเป็นครูจาก FreeGeek.org, นักนำเสนอที่
DesertCodeCamp, ผู้ใช้ Wikipedia and สมาชิก LinuxChix.
เธอจัดการและประชาสัมพันธ์การศึกษาความปลอดภัยบนลินุกซ์ ไปยัง Phoenix Linux Users
Group HackFEST Series labs, ใช้สองเสาร์ของทุก ๆ เดือนที่ The
Foundation for Blind Children in Phoenix, Arizona. Obnosis.com, a play
on a words coined by LRHubbard, ลงทะเบียนใน in the 1990's, เป็น "word
hack" จาก the Church of Scientology, หลังจาก 6 ปีของผู้ดูแลข่าว UseNet.
ความภูมิใจที่สุดของคือคือการที่ได้นั้งกับ Linux Torvald's
ระหว่างการสัมภาษณ์ที่ OSDL.org ใน Oregon ในปี 2002.



ตีพิมพ์ในเล่มที่ 165 ของ of Linux Gazette, สิงหาคม 2009

- 
- อ่าน 5214 ครั้ง



**Links**

[1] http://linuxgazette.net/authors/kachold.html
[2] http://sake.in.th
[3] http://ostatic.com/facter-ruby
[4] http://rubygems.org/read/book/2
[5] http://reductivelabs.com/trac/puppet/wiki/InstallationGuide#InstallPuppet
[6] http://reductivelabs.com/trac/puppet/wiki/Recipes/OpenNTPD
[7] http://reductivelabs.com/trac/puppet/wiki/Recipes/FilePermissionCheck
[8] http://reductivelabs.com/trac/puppet/wiki/Recipes/Sudo

[9] http://reductivelabs.com/trac/puppet/wiki/Recipes/CentralizedSudoers
[10] http://reductivelabs.com/trac/puppet/wiki/Recipes/AptKeys
[11] http://reductivelabs.com/trac/puppet/wiki/Recipes/ModuleIptables
[12] http://reductivelabs.com/trac/puppet/wiki/Recipes/AqueosShorewall
[13] http://reductivelabs.com/trac/puppet/wiki/Recipes/sshdconfigTemplate
[14] http://reductivelabs.com/trac/puppet/wiki/Recipes/Nagios
[15] http://reductivelabs.com/trac/puppet/wiki/Recipes/Authorized_keys
[16] http://reductivelabs.com/trac/puppet/wiki/Recipes/ClamAV
[17] http://reductivelabs.com/trac/puppet/wiki/Recipes/UserAndHomedirRecipe
[18] http://reductivelabs.com/trac/puppet/wiki/Recipes/PasswordManagement
[19] http://reductivelabs.com/trac/puppet/wiki/Recipes/FirmwarePassword
[20] http://reductivelabs.com/trac/puppet/wiki/Recipes/YumServerBuild
[21] http://reductivelabs.com/trac/puppet/wiki/Recipes/ResolvConf
[22] http://reductivelabs.com/trac/puppet/wiki/Recipes/Solaris_cde-login
[23] http://reductivelabs.com/trac/puppet/wiki/Recipes/ZabbixAgent
[24] http://reductivelabs.com/trac/puppet/wiki/SimplestPuppetInstallRecipe
[25] http://reductivelabs.com/trac/puppet/wiki/Recipes/SimpleText
[26] http://plug.phoenix.az.us/node/659
[27] http://plug.obnosis.com/
[28] mailto:tag@lists.linuxgazette.net?subject=Talkback:165/kachold.html
[29] http://linuxgazette.net/copying.html