Security Hardening F5's BIG-IP 12.0 with SELinux

Chase Abbott, 2015-15-09

When a major release hits the street, documentation and digital press tends to focus on new or improved user features, seldom do underlying platform changes make the spotlight. BIG-IP 12.0 has plenty of new customer-centric features but one unsung massive ***feature*** is SELinux's extensive enforcing mode policy across the architecture. This isn't one change nor is this one group making a few code updates. This represents a commitment by product development to ensure their code and modules comply with the ever-increasing security requirements we demand of our platforms. We practice what we preach; F5 is a security company. BIG-IP and SELinux are no strangers, having coexisted since 2009, but comparing previous policies to 12.0 is akin to comparing a domestic house cat to a Sabre Toothed Tiger.

A Brief Understanding of SELinux

In Linux-land, security is enforced via Discretionary Access Controls (DAC). A resource is associated with a user/group and given a permission set (read, write, execute). The permissions of a resource are validated against the requestor and granted or denied based on those permissions. This is useful for user access to resources but most processes run root level permissions; you know, the account you're not supposed to run things as if you don't have to. An application or user running as root, if compromised, negates the security defined on the system.

SELinux (Security-Enhanced Linux) provides secondary granular security complimenting DAC with Mandatory Access Control (MAC) policies. SELinux implements a policy database to govern how a subject is allowed to interact with an object, and either grants permission or denies access based on those predefined rules. A subject in this context is any process that "acts" on something; an object is whatever is being acted on by the subject. This could be an http daemon (subject) requesting a port (object) or mysql (subject) accessing it's own database files (object). Since an object can be a directory, file, socket, pipe, memory, IPC… the amount of potential policy rules one can create on a system is overwhelming. Now you're starting to realize how large this undertaking was by our PD and I still oversimplified a great deal of the complexity (sorry John).

BIG-IP 12.0 and SELinux

SELinux can run in Permissive or Enforcing mode, where permissive will log denials but still allow the interaction of subject and object (disabled is also a mode, albeit not useful in our case). Enforcing mode will log and prohibit any policy violations and is how security appliances should work. To run sestatus and see `Current mode: enforcing` is only step A. If the SELinux policy is empty, you're enforcing nothing. To get a better idea of the massive policy expansion between 11.6 and 12.0, we can compare policy summary on each version (policy.21 versus policy.24):

v.11.6

```
 Reading policy...
libsepol.policydb_index_others: security: 3 users, 6 roles, 1904 types, 258 bools
libsepol.policydb_index_others: security: 1 sens, 1024 cats
libsepol.policydb_index_others: security: 65 classes, 105769 rules, 53457 cond rules
```

```
binary policy file loaded
```

v.12.0

```
 Reading policy...
libsepol.policydb_index_others: security: 9 users, 12 roles, 3974 types, 202 bools
libsepol.policydb_index_others: security: 1 sens, 1024 cats
libsepol.policydb_index_others: security: 81 classes, 359039 rules, 275855 cond rules
binary policy file loaded
```

v.12.1

```
Reading policy...
libsepol.policydb_index_others: security:  9 users, 12 roles, 4053 types, 225 bools
libsepol.policydb_index_others: security: 1 sens, 1024 cats
libsepol.policydb_index_others: security:  81 classes, 361831 rules, 305448 cond rules
binary policy file loaded
```

v.12.1.1 HF2

```
Reading policy...
libsepol.policydb_index_others: security:  9 users, 12 roles, 4054 types, 225 bools
libsepol.policydb_index_others: security: 1 sens, 1024 cats
libsepol.policydb_index_others: security:  81 classes, 361933 rules, 305482 cond rules
binary policy file loaded
```

Note the growth in rules and in conditional rules; product development has been very busy to ensure functional parity while further restricting the permission maps of how processes operate in their respective domains against their intended targets. Any attempt to override or elevate privileges on processes or objects would be met with a denial and log entry.

Now what….

SELinux on BIG-IP is one of those features you should be super excited about but don't actually play with, and that's a good thing. For the SELinux admin, you understand the effort required and it's quite impressive. For those new to SELinux and think this isn't a big deal, there are plenty of information on the internet to help understand how it can improve and complicate your various systems. In BIG-IP we took care of the heavy lifting for you. Below are further resources for further reading so you can be the life of any party. Thanks for playing.

Related SELinux Resources:

The Debian Administrator's Handbook: Introduction to SELinux

CentOS Howtos: SELinux

Dan Walsh's Blog

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com