목차

SetUID, SetGID, Sticky Bit 3
먼저보기 3
1. 공유 클래스 옵션 3
**** 가이드 받은내용*** 3
SetUID, SetGID, Sticky Bit 란? 4
1. SetUID 4
2. SetGID 4
3. Sticky Bit 4
설정예제 5
디렉토리 퍼미션 777 (drwxrwxrwx) 5
디렉토리 퍼미션 700 (drwx------) 5
디렉토리 퍼미션 1777 (drwxrwxrwt) <- 공유 폴더 (/tmp 와 동일) (Sticky Bit) 5

SetUID, SetGID, Sticky Bit

출처 : http://soul0.tistory.com/246

SetUID,SetGID,Sticky Bit 란? 권한이 어떻게 될까?

먼저보기

SetUID와 SetGID, Sticky Bit를 살펴보게 된 계기는 아래와 같습니다. Other 권한을 없애햐 하는데, AIX, LINUX 에서 /tmp 파일시스템 자체가 공유를 위한 목적이다 보니 Other 에 rwx 권한 모두 있게 되었는데요. 보안검사때문에 Other 에서 권한을 빼야하는 경우입니다. 실제로, WebSphere 부분에서 /tmp/javasharedresources 를 사용하고 있구요.. WAS 에서 컴파일할때 캐시목적으로 쓰는 곳입니다. 아래 쓰여져 있는 말을 인용하면

1. 공유 클래스 옵션

- 캐시에서 클래스를 공유하면 시작시간이 개선 - Instance, NodeAgnet, DMGR 에서는 공유 클래스 옵션을 사용할 수 있음 - 사용중이 아닐때에는 삭제 - 관리콘솔에서 JVM Option 에 `-Xshareclasses:node` 을 이용하여 생성하지 않도록 변경 가능 (권장하지 않음)

8. **캐시에서 클래스 공유를 사용 가능하게 하십시오.**

IBM J2RE(Java 2 Runtime Environment) 버전 1.5.0의 공유 클래스 옵션은 캐시에서 클래스를 공유할 수 있게 합니다. 캐시에서 클래스를 공유하면 시작 시간이 개선되고 메모리 풋프린트가 감소합니다. Application Server, Node Agent 및 Deployment Manager와 같은 프로세스는 공유 클래스 옵션을 사용할 수 있습니다.

이 옵션을 사용하면 프로세스가 사용 중이지 않을 때 캐시를 지워야 합니다. 캐시를 지우려면 *app_server_root*`/bin/clearClassCache.bat/sh` 유틸리티를 호출하거나 프로세스를 중지한 후 프로세스를 다시 시작하십시오.

프로세스에 대한 공유 클래스 옵션을 사용 안해야 하는 경우 해당 프로세스에 대해 일반 JVM 인수 -Xshareclasses:none을 지정하십시오.

a. 관리 콘솔에서, **서버 > Application Servers >** *server*를 클릭하십시오.
b. 서버 인프라 아래에서 **Java 및 프로세스 관리 > 프로세스 정의**를 클릭하십시오.
c. **제어** 또는 **하위**를 선택한 다음 **JVM(Java Virtual Machine)**을 선택하십시오.
d. 일반 JVM 인수 필드에서 `-Xshareclasses:none`를 입력하십시오.
e. **확인**을 클릭하십시오.
f. 마스터 구성에 변경사항을 저장하십시오.
g. Application Server를 중지했다가 다시 시작하십시오.

기본값:	캐시 옵션의 공유 클래스가 사용 가능합니다.
권장:	캐시에서 클래스 공유 옵션을 사용 가능으로 두십시오.
사용법:	-Xshareclasses:none은 캐시에서 클래스 공유 옵션을 사용 불가능하게 합니다.

* 가이드 받은내용

1. `/tmp/javasharedresources` 안에 생성된 파일들의 권한을 소유권한을 check 하여, `javasharedresources` 디렉토리 자체를 소유권한 변경 및 permission 설정을 고려해볼수 있을 것 같네요..
1. `-Xsharedclasses` 옵션에 추가하여 다른 디렉토리에 쌓이게 하는 방법도 있을수 있겠네요 : `controlDir=<your_was_home>/tmp` 그러나 해당 디렉토리 또한 777 입니다. 1번항목이 적절해 보입니다.
1. disable 하는것은 권장사항이 아닙니다.(none)

WAS 가 OS 의 /tmp 를 사용하는 고정공간으로는 `/tmp/.com_ibm_tools_attach/` 와 `/tmp/javasharedresources/` 공간입니다. 하기에서 언급된 `/tmp/javasharedresources` 는 java class cache 영역으로, 해당디렉토리 밑에 별도의 cache 파일이 생성됩니다. 그것은 Version 6.1 부

터 도입된 기능으로, 권장사항은 사용하도록 하는것입니다(default 값). 말그대로 shared resource 이기 때문에, javasharedresources 디렉토리가 777 로 디렉토리 생성될때 설정이 됩니다. WAS 에서 사용하는 umask 값은 022 입니다..따라서 해당 directory 안에 생성되는 파일은 당연히 644 로 생성되게 됩니다. 근데 태생이 shared 되는 resource 로써, shared 하기 위해 만들어지는 영역이 자체가 문제가라는 건지, 이부분이 이해가 되질 않네요. 공유자체가 보안risk 가 있다(?)고 해서 하는것인지요? (왜 취약한지 그것이 문제네요..이게 문제라면 태생적 공유를 목적으로 탄생한 /tmp 디렉토리 아예없어야져야 할듯..그러나 이건 아니기에)

아무튼 권장은 사용입니다.

SetUID, SetGID, Sticky Bit 란?

파일의 권한은 일반적으로 사용자권한, 그룹권한, 아더권한이 있고, 각각의 권한에 대해 읽기권한®, 쓰기권한(w), 그리고 실행권한(x)가 있다. 근데 사용자권한, 그룹권한, 아더권한 외에도 특수권한 이라는게 있다. 특수 권한을 이용하면 다른 계정에게 읽기,쓰기,실행 권한을 줄 수 있으며, 다른 계정(A)에서 만든 파일을 내 계정(B)에서 실행하는동안 A의 권한을 얻어 해당 파일을 읽거나 쓰거나 실행하는 것이 가능해진다 (SetUID와 SetGID에 해당). 특수 권한에는 `SetUID`, `SetGID`, `Sticky Bit`가 있다.

1. SetUID

사용자의 권한이 있어야만 실행을 할 수 있는 파일의 경우, 그 권한을 일시적으로 파일을 실행하는 일반 사용자들에게 부여하기 위해 사용 한다. SetUID가 설정된 파일을 실행하면 해당 파일을 실행하는 도중에는 파일의 사용자 권한을 획득하며, 파일의 실행을 마친 뒤에는 다시 자신의 권한으로 돌아온다.

SetUID는 4xxx로 설정한다. (100에 해당. xxx는 기존의 사용자권한, 그룹권한, 아더권한을 의미)
SetUID 설정 시 사용자 권한내의 실행 권한 자리에 x가 아닌 s가 오게 된다. ex) chmod 4750 httpd (rwsr-x—)

2. SetGID

그룹의 권한이 있어야만 실행을 할 수 있는 파일의 경우, 그 권한을 일시적으로 파일을 실행하는 일반 사용자들에게 부여하기 위해 사용 한다. SetGID가 설정된 파일을 실행하면 해당 파일을 실행하는 도중에는 파일의 그룹 권한을 획득하며, 파일의 실행을 마친 뒤에는 다시 자신의 권한으로 돌아온다.

SetGID는 2xxx로 설정한다. (010에 해당. xxx는 기존의 사용자권한, 그룹권한, 아더권한을 의미)
SetGID 설정 시 그룹 권한내의 실행 권한 자리에 x가 아닌 s가 오게 된다. ex) chmod 2750 httpd (rwxr-s—)

3. Sticky Bit

Sticky Bit가 설정된 디렉토리에 파일을 생성하면 해당 파일은 생성한 사람의 소유가 되며, 오직 소유자와 root에게만 해당 파일에 대한 삭제 및 변경의 권한이 있다. 공유디렉토리로 사용하고자 할 때 쓰인다.

Sticky Bit는 1xxx로 설정한다. (001에 해당. xxx는 기존의 사용자권한, 그룹권한, 아더권한을 의미)
Sticky Bit 설정 시 아더 권한내의 실행 권한 자리에 x가 아닌 t가 오게 된다.

설정예제

디렉토리 퍼미션 777 (drwxrwxrwx)

- 해당 폴더안에 생성된 파일의 소유자 및 모든 사람이 삭제할수 있다.
- root 가 만든 파일의 퍼미션이 600 (-rw——-) 이더라도, 디렉토리 퍼미션 777 (drwxrwxrwx) 아래 있으면 일반계정에서 삭제할수 있다.

디렉토리 퍼미션 700 (drwx------)

- 해당 폴더 소유자는 삭제관점에선 전권을 가지고 있다.
- root 가 만든 파일의 퍼미션이 600 (-rw——-) 이더라도, 디렉토리 퍼미션 700 (drwx——) 아래 있으면 해당 폴더 소유자는 삭제할수 있다

디렉토리 퍼미션 1777 (drwxrwxrwt) <- 공유 폴더 (/tmp 와 동일) (Sticky Bit)

- 해당 폴더안에 생성된 파일의 소유자와 root 만이 삭제할수 있다.
- 해당 폴더안에 파일의 퍼미션이 777 (-rwxrwxrwx) 이러라도, 디렉토리 퍼미션 1777 (drwxrwxrwt) 아래 있으면 파일의 소유자 및 root 만이 삭제할수 있다. 즉, root 및 파일의 owner 만 삭제할수 있습니다.

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/setuid_setgid_sticky_bit

Last update: **2018/11/18 15:53**