```
.h
class zbl_Building_Attack_Announce : public ScriptImpClass
{
 bool announced;
 int team;
public:
 void Created(GameObject *o);
 void Damaged(GameObject *o, GameObject *d, float damage);
 void Timer_Expired(GameObject *o, int num);
};

.cpp

void zbl_Building_Attack_Announce::Created(GameObject *o)
{
 announced = 0;
 team = Get_Int_Parameter("Object_Team");
}
void zbl_Building_Attack_Announce::Damaged(GameObject *o, GameObject *d, float damage)
{
 if (!announced)
 {
  unsigned int r = 255, g = 255, b = 255;
  char Announce[256];
  sprintf(Announce, "Warning - The %s is under attack!", Get_Parameter("Translated"));
  Send_Message_Team(team, r, g, b, Announce);
  Create_2D_Sound_Team(Get_Parameter("Sound"), team);
  announced = 1;
  Commands->Start_Timer(o, this, Get_Float_Parameter("Interval"), 123);
 }
}
void zbl_Building_Attack_Announce::Timer_Expired(GameObject *o, int num)
{
 if (num == 123)
 {
  if (announced)
  {
   announced = 0;
  }
 }
}
```

Toggle SpoilerScriptRegistrant<zbl_Building_Attack_Announce>
zbl_Building_Attack_Announce_Registrant("zbl_Building_Attack_Announce",

"Object_Team=0:int,Interval=30000.00:float,Sound=amb_airraid.wav:string", "Translated=GDI Weapons Factory:string");

Parameters:

Object_Team:int - The team the object belongs to.
Interval:float - The interval at which to announce "<building> is under attack!"
Sound:string - The sound file to play for the team, e.g. "Warning - Nod Airstrip Under Attack!" etc
Translated:string - The Translated Screenname of the object (what you want the object to be called)

The "Translated" parameter is there because I didn't want to look for a translation function nor include one for getting the preset's screenname. If another coder finds one, you're more than welcome to replace this parameter with that function call.