



GOLIATH NATIONAL BANK

# SECURE CODING

TEAM 12 - PHASE 3

MEMBERS

Alexander Lill    Lorenzo Donini

---

Florian Mauracher    Mahmoud Naser

# USE-CASES



# USE-CASES PHASE 1

Usecase	Status
Customer / Employee registration (including sending e-mail with TANs)	WORKING
Customer / Employee login	WORKING
Customer / Employee logout	WORKING
Customer / Employee views bank account details of Customer	WORKING
Customer / Employee views transaction history of Customer	WORKING
Customer money transfer via HTML form (using TAN)	WORKING
Customer money transfer via uploading transaction batch file (using TAN)	WORKING
Employee approves transfers larger than 10.000 EUR	WORKING
Employee approves registration of Customer or of other employee	WORKING
Customer / Employee downloads transaction history of Customer as PDF	WORKING
Transaction verification page	WORKING
Search feature for user accounts	WORKING
Employee rejects transfers larger than 10.000 EUR	WORKING
Employee rejects registration of Customer or of other employee	WORKING



# USE-CASES PHASE 3

Usecase	Status
Password recovery	WORKING
Encrypted TAN generation and delivery via e-mail to customer	WORKING
Download of SCS after registration	WORKING
Transfer using SCS (Single transaction)	WORKING
Transfer using SCS (Batch transaction)	WORKING



# VULNERABILITIES

# THE HTTPS MIGRATION

Moving the website from HTTP to HTTPS solved a large amount of issues that was related to clear text data.

The image consists of two side-by-side screenshots from the video game DOOM. The left screenshot shows a dark, stone-walled corridor with a player character visible in the distance. The right screenshot shows the same scene but with significantly improved textures and lighting, making the environment look more detailed and modern. This visual comparison highlights the significant improvements in texture quality and rendering that occurred during the migration from HTTP to HTTPS.

# MOVING THE WEB SERVER DOCUMENT ROOT

One of the major vulnerabilities was the fact that a few files were accessible through the webserver DocumentRoot. These files included sensitive information regarding the configuration of the system. Our solution was to change the webserver DocumentRoot to the folder that only contains the PHP code and no sensitive information.

One of the major vulnerabilities was the fact that a few files were accessible through the webserver DocumentRoot. These files included sensitive information regarding the configuration of the system. Our solution was to change the webserver DocumentRoot to the folder that only contains the PHP code and no sensitive information.

One of the vulnerabilities was a timing attack, where the login page would reply within 50ms if the mail address was valid and within around 25ms if the mail address was not valid. While minor, this attack could be used to check whether a certain email is registered at our bank or not. To overcome this a random customized waiting period was introduced to all login attempts, making the response time for both cases very similar.

## TIMING ATTACK

```

if ((gameremode == commercial) && (SPR_BON2_2,0,(NULL),S_BON2,0,0)) // secretexit = false;
  || (gameremode == pack_trc) && (SPR_BON2_1,0,(NULL),S_BON2,0,0)) // S_BON2
  || (gameremode == pack_plut) && (SPR_BON2_0,0,(NULL),S_BKEY2,0,0)) // S_SKY2
  || (gameremode == pack_pvt) && (SPR_BON2_3,0,(NULL),S_BKEY2,0,0)) // S_SKY2
skytexture = R_TextureNumForName ("SKY3");
if (gamermap < 12)
  skytexture = R_TextureNumForName ("SKY1");
else
  if (gamermap < 21)
    skytexture = R_TextureNumForName ("SKY2");
  else
    skytexture = R_TextureNumForName ("SKY1");
}

levelstartic = gametic; // for time calculation
if (wipegamestate == GS_LEVEL)
  wipegamestate = -1; // force a wipe
gamerate = GS_LEVEL;
for (i=0 ; i<MAXPLAYERS ; i++)
{
  if (playeringameli && !gameremode) // gamemode != commercial
    playersli.playerstate = PST_DEAD;
  memset (playersli frags, 0, sizeof(playersli frags));
}
P_SetupLevel (gameepisode, gamermap, 0, gamekill);
displayplayer = consoleplayer;
starttime = _GetTime();
gameaction = ga_nothing;
Z_CheckHeap();
// clear cmd building stuff
memset (gamekeydown, 0, sizeof(gamekeydown));
joyxmove = joyymove = 0;
mousex = mousey = 0;
sendpause = sendsave = paused = false;
memset (mousebuttons, 0, sizeof(mousebuttons));
memset (joybuttons, 0, sizeof(joybuttons));
void G_PlayerReborn (int player)
{
  player_t* p;
  int i;
  fragsMAXPLAYERS;
  killcount;
  itemcount;
  secretcount;
  wminfo didsecret = players[consoleplayer].didsecret;
  killcount = players[player].killcount;
  itemcount = players[player].itemcount;
  secretcount = players[player].secretcount;
  p = &players[player];
  memset (p, 0, sizeof(*p));
  if (gamermode == commercial)
    memcpy (players[player].frags, frags, sizeof(frags));
  players[player].killcount = killcount;
  players[player].itemcount = itemcount;
  players[player].secretcount = secretcount;
  p->usedown = p->attackdown = true;
  p->playerstate = PST_LIVE;
  p->health = MAXHEALTH;
  p->readyweapon = p->pendingweapon = wp_pistol;
  p->weaponowned[wlp_fist] = true;
  p->weaponowned[wlp_pistol] = true;
  p->ammo/am_clip1 = 50;
  p->ammo/am_clip2 = 50;
  for (i=0 ; i<NUMAMMO ; i++)
    p->maxammo[i] = maxammo;
  if (secretexit)
    mprintf ("SECRET %d\n", secretexit);
  for (i=0 ; i<NUMHEAD ; i++)
    p->maxammo[i] = maxammo;
  if (secretexit)
    mprintf ("SECRET %d\n", secretexit);
  P_SpawnPlayer (mapthing_t* mthing);
  boolean G_CheckSpot (playermem_t* playermem, mapthing_t* mthing);
  fixed_t x;
  fixed_t y;
  subsector_t* ss;
  unsigned an;
  mobj_t* mo;
  int i;
  length = M_ReadFile (savename, &savebuffer);
  save_p = savebuffer + SAVESTRINGSIZE;
  if (paused)
  {
    paused = false;
    S_ResumeSound (0);
  }
  if (skill > sk_nightmare)
    skill = sk_nightmare;
  if (paused)
  {
    if (paused)
      paused = false;
    S_Pistol (0,1,(A_Raise),S_PISTOLUP,0,0); // S_PISTOLUP
    (SPR_PISG_0,1,(A_Raise),S_PISTOLUP,0,0); // S_PISTOL1
    (SPR_PISG_1,0,(NULL),S_PISTOL2,0,0); // S_PISTOL2
    (SPR_PISG_2,0,(NULL),S_PISTOL4,0,0); // S_PISTOL3
    (SPR_PISG_1,5,(A_Refire),S_PISTOL,0,0); // S_PISTOL4
    (SPR_PISF_32768,7,(A_Light1),S_LIGHTDONE,0,0); // S_PISTOLFLASH
    (SPR_SHTG_0,1,(A_WeaponReady),S_SGUN,0,0); // S_SGUN
    (SPR_SHTG_0,1,(A_Lower),S_SGUNDOWN,0,0); // S_SGUNDOWN
    (SPR_SHTG_0,1,(A_Raise),S_SGUNUP,0,0); // S_SGUNUP
    (SPR_SHTG_0,3,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,0,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,10,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,20,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,30,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,40,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,50,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,60,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,70,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,80,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,90,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,100,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,110,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,120,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,130,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,140,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,150,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,160,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,170,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,180,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,190,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,200,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,210,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,220,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,230,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,240,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,250,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,260,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,270,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,280,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,290,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,300,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,310,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,320,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,330,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,340,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,350,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,360,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,370,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,380,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,390,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,400,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,410,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,420,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,430,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,440,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,450,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,460,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,470,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,480,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,490,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,500,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,510,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,520,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,530,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,540,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,550,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,560,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,570,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,580,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,590,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,600,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,610,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,620,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,630,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,640,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,650,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,660,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,670,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,680,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,690,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,700,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,710,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,720,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,730,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,740,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,750,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,760,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,770,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,780,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,790,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,800,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,810,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,820,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,830,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,840,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,850,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,860,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,870,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,880,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,890,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,900,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,910,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,920,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,930,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,940,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,950,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,960,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,970,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,980,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,990,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1000,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1010,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1020,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1030,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1040,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1050,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1060,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1070,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1080,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1090,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1100,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1110,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1120,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1130,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1140,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1150,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1160,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1170,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1180,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1190,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1200,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1210,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1220,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1230,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1240,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1250,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1260,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1270,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1280,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1290,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1300,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1310,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1320,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1330,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1340,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1350,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1360,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1370,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1380,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1390,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1400,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1410,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1420,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1430,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1440,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1450,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1460,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1470,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1480,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1490,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1500,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1510,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1520,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1530,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1540,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1550,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1560,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1570,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1580,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1590,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1600,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1610,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1620,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1630,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1640,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1650,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1660,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1670,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1680,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1690,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1700,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1710,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1720,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1730,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1740,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1750,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1760,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1770,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1780,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1790,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1800,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1810,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1820,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1830,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1840,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1850,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1860,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1870,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1880,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1890,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1900,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1910,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1920,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1930,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1940,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1950,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1960,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1970,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1980,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,1990,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2000,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2010,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2020,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2030,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2040,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2050,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2060,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2070,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2080,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2090,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2100,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2110,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2120,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2130,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2140,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2150,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2160,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2170,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2180,(NULL),S_SKULL2,0,0); // S_SKULL2
    (SPR_SKULL,2190,(NULL),S_SK
```

**UNSANITIZED INPUT**

# Creating a unified sanitizing function was the solution to combating unsanitization.

# UNSANITIZED INPUT

Creating a unified sanitizing function was the solution to combating unsanitization.

```
if (gamemode == commercial) {  
    SPR_BON2,2,6,(NULL),S_BON2,0,0, // S_BON2  
    SPR_BON2,1,6,(NULL),S_BON2,0,0, // S_BON2  
    SPR_BKEY,0,10,(NULL),S_BKEY2,0,0, // S_SKY2  
    gameaction = ga_completed;  
    skytexture = R_TextureNumForName ("SKY3");  
    if (gamemap < 12) {  
        skytexture = R_TextureNumForName ("SKY1");  
    } else {  
        if (gamemap < 21) {  
            skytexture = R_TextureNumForName ("SKY2");  
        } else {  
            skytexture = R_TextureNumForName ("SKY1");  
        }  
    }  
}  
  
levelstartic = gametic; // for time calculation  
  
if (wipegamestate == GS_LEVEL) {  
    wipegamestate = -1; // force a wipe  
}  
  
gamestate = GS_LEVEL;  
  
for (i=0 ; i<MAXPLAYERS ; i++) {  
    if (playeringame[i] && players[i].playerstate == PST_DEAD) {  
        players[i].playerstate = PST_REBORN;  
        memset (players[i]. frags, 0, sizeof(players[i]. frags));  
    }  
}  
  
P_SetupLevel (gameepisode, gamemap, 0, gamekill);  
displayplayer = consoleplayer;  
starttime = _GetTime ();  
gameaction = ga_nothing;  
Z_CheckHeap ();  
  
// clear cmd building stuff  
memset (gamekeydown, 0, sizeof(gamekeydown));  
joymove = joymove = 0;  
mousex = mousey = 0;  
sendpulse = sendpulse = paused = false;  
memset (mousebuttons, 0, sizeof(mousebuttons));  
memset (joybuttons, 0, sizeof(joybuttons));  
  
void G_PlayerReborn (int player)  
{  
    player_t* p;  
    int i;  
    fragsMAXPLAYERS;  
    killcount;  
    itemcount;  
    secretcount;  
  
    wminfo didsecret = players[consoleplayer].didsecret;  
    killcount = players[player].killcount;  
    itemcount = players[player].itemcount;  
    secretcount = players[player].secretcount;  
  
    p = &players[player];  
    memset (p, 0, sizeof(*p));  
  
    memcpy (players[player].frags, frags, sizeof(frags));  
    players[player].killcount = killcount;  
    players[player].itemcount = itemcount;  
    players[player].secretcount = secretcount;  
  
    p->usedown = p->attackdown = true;  
    p->playerstate = PST_LIVE;  
    p->health = MAXHEALTH;  
    p->readyweapon = p->pendingweapon = wp_pistol;  
    p->weaponowned[wlp_fist] = true;  
    p->weaponowned[wlp_pistol] = true;  
    p->ammo(am_clip) = 50;  
    p->pol(4,0,(NULL),S_HEADCANDLES);  
    for (i=0 ; i<NUMAMMO ; i++) {  
        p->maxammo[i] = maxammo;  
    }  
  
    void P_SpawnPlayer (mapthing_t* mobj_t, playernum_t playernum, mapthing_t* mthing_t, fixed_t x, fixed_t y, subsector_t* ss, unsigned an, mobi_t* mo, int i);  
  
    if (gamemode == commercial) {  
        if (gamemode == pack_trc) {  
            skytexture = R_TextureNumForName ("SKY3");  
        } else {  
            if (gamemode == pack_plt) {  
                skytexture = R_TextureNumForName ("SKY1");  
            } else {  
                if (gamemode == pack_cpt) {  
                    skytexture = R_TextureNumForName ("SKY2");  
                } else {  
                    skytexture = R_TextureNumForName ("SKY1");  
                }  
            }  
        }  
    }  
  
    length = M_ReadFile (savename, &savebuffer);  
    save_p = savebuffer + SAVESTRINGSIZE;  
  
    // skip the description field  
    memset (&check, 0, sizeof(&check));  
    sprint (&check, "version %i", VERSION);  
    if (strcmp (&save_p, &check)) {  
        return; // bad version  
    }  
    save_p += VERSIONSIZE;  
  
    gameskill = *save_p++;  
    gameepisode = *save_p++;  
    gamemap = *save_p++;  
    for (i=0 ; i<MAXPLAYERS ; i++) {  
        playeringame[i] = *save_p++;  
    }  
  
    // load a base level  
    G_InitNew (gameskill, gameepisode, gamemap);  
  
    // get the times  
    a = *save_p++;  
    b = *save_p++;  
    c = *save_p++;  
    levetime = (a<<16) + (b<<8) + c;  
  
    // dearchive all the modifications  
    P_UnArchivePlayers ();  
    P_UnArchiveWorld ();  
    P_UnArchiveThinkers ();  
    P_UnArchiveSpecials ();  
  
    if (*save_p != 0xd) {  
        _Error ("Bad save game");  
    }  
  
    // done  
    Z_Free (savebuffer);  
  
    if (setsizeneeded) {  
        R_ExecuteSetViewSize ();  
    }  
  
    // draw the pattern into the back screen  
    R_FillBackScreen ();  
    R_DrawBackPattern ();  
  
    void G_DoSaveGame (void)  
{  
    char name[100];  
    char name2[VERSIONSIZE];  
    char* description;  
    int length;  
    int i;  
  
    if (fastparm) {  
        if (skill == sk_nightmare || respawnparm) {  
            respawmmonsters = true;  
        } else {  
            respawmmonsters = false;  
        }  
    }  
  
    if (M_CheckParm ("-drom")) {  
        sprintf (name, "c:\\doomdata\\%s\\SAVEGAMENAME%id.dsg", savegameslot);  
    } else {  
        sprintf (name, "%s\\SAVEGAMENAME%id.dsg", savegameslot);  
    }  
  
    description = savedescription;  
  
    save_p = savebuffer + screens[1]+0x400;  
  
    for (i=S_SARG_RUN1 ; i<=S_SARG_PAIN2 ; i++) {  
        states[i].tics >= 1;  
        mobinfo[MT_BRUISERSHOT1].speed = 20*FRACUNIT;  
        mobinfo[MT_HEADSHOT1].speed = 20*FRACUNIT;  
        mobinfo[MT_TROOPSHOT1].speed = 20*FRACUNIT;  
    }  
  
    if (skill == sk_nightmare && gamekill == sk_nightmare) {  
        for (i=S_SAWG_RUN1 ; i<=S_SAWG_PAIN2 ; i++) {  
            states[i].tics <= 1;  
            mobinfo[MT_BRUISERSHOT1].speed = 15*FRACUNIT;  
            mobinfo[MT_HEADSHOT1].speed = 10*FRACUNIT;  
            mobinfo[MT_TROOPSHOT1].speed = 10*FRACUNIT;  
        }  
    }  
  
    if (skill == sk_nightmare && gamekill == sk_nightmare) {  
        for (i=S_SAWG_RUN1 ; i<=S_SAWG_PAIN2 ; i++) {  
            states[i].tics <= 1;  
            mobinfo[MT_BRUISERSHOT1].speed = 15*FRACUNIT;  
            mobinfo[MT_HEADSHOT1].speed = 10*FRACUNIT;  
            mobinfo[MT_TROOPSHOT1].speed = 10*FRACUNIT;  
        }  
    }  
  
    // force players to be initialized upon first level load  
    for (i=0 ; i<MAXPLAYERS ; i++) {  
        players[i].playerstate = PST_REBORN;  
    }  
  
    username = true; // will be set false if a demo  
    paused = false;  
    demoplayback = false;  
    autowepactive = false;  
    viewactive = true;  
    gameepisode = episode;  
    gamemap = map;  
    gameskill = skill;  
  
    viewactive = true;  
  
    // set the sky map for the episode  
    if (gamemode == commercial) {  
        skytexture = R_TextureNumForName ("SKY3");  
        if (length > SAVEGAMEFSIZE) {  
            _Error ("Savegame buffer overrun");  
        }  
        if (gamemap < 12) {  
            skytexture = R_TextureNumForName ("SKY1");  
        } else {  
            skytexture = R_TextureNumForName ("SKY2");  
        }  
    }  
  
    if (paused) {  
        if (ga_paused) {  
            paused = false;  
            S_ResumeSound ();  
        }  
    }  
  
    if (skill > sk_nightmare) {  
        skill = sk_nightmare;  
    }  
  
    // This was quite messy with SPECIAL and commented parts.  
    // Supposedly hacks to make the latest edition work.  
    // It might not work properly.  
    if (episode < 1) {  
        episode = 1;  
    }  
  
    if (gamemode == retail) {  
        if (episode > 4) {  
            episode = 4;  
        }  
        else if (gamemode == shareware) {  
            if (episode > 1) {  
                episode = 1; // only start episode 1 on shareware  
            }  
        }  
    }  
  
    if (sec->validcount == validcount) {  
        if (sec->soundtraversed == soundblocks+1) {  
            return; // already flooded  
        }  
    }  
  
    if (sec->validcount = validcount);  
    sec->soundtraversed = soundblocks+1;  
    sec->soundtarget = soundtarget;  
  
    for (i=0 ; i<sec->linecount ; i++) {  
        check = sec->lines[i];  
        if (! (check->flags & ML_TWOSIDED)) {  
            continue;  
        }  
        P_LineOpening (check);  
  
        if (openrange <= 0) {  
            continue; // closed door  
        }  
  
        if (sidesel->sidenum[0] != sector->sector) {  
            other = sidesel->sidenum[1] != sector->sector;  
        } else {  
            other = sidesel->sidenum[0] != sector->sector;  
        }  
  
        if (check->flags & ML_SOUNDBLOCK) {  
            if (!soundblocks) {  
                P_RecursiveSound (other, 1);  
            } else {  
                P_RecursiveSound (other, soundblocks);  
            }  
        }  
  
        static char turbomessage[80];  
        extern char *player_names[4];  
        sprintf (turbomessage, "%s is turbo", player_names[i]);  
        players[consoleplayer].message = turbomessage;  
  
        if (netgame && initedemo && !(gematic%ticdup)) {  
            // If a monster yells at a player;  
            // it will alert other monsters to the player.  
            if (gematic > BACKUPTICS && consistency(lbuf) != cmd->consistency) {  
                void P_NoiseAlert ();  
            }  
            L_Error ("consistency failure (%i should be %i)", cmd->consistency, consistency(lbuf));  
            target = emitter; // target, emitter )  
        }  
  
        if (players[i].mo) {  
            soundtarget = target;  
            consistency(lbuf) = players[i].mo->x;  
        } else {  
            consistency(lbuf) = rndindex;  
        }  
  
        if (playeringame[i]) {  
            boolean P_CheckMeleeRange (mobi_t* actor, mobi_t* pl, fixed_t dist);  
            if (players[i].cmd.buttons & BT_SPECIAL) {  
                switch (players[i].cmd.buttons & BT_SPECIALMASK) {  
                    case BTS_PAUSE: {  
                        if (factor > target) {  
                            paused ^= 1;  
                        } if (paused) {  
                            S_PauseSound ();  
                        } else {  
                            S_ResumeSound ();  
                        }  
                    } break;  
                }  
            }  
        }  
    }  
}
```

# INTEGER OVERFLOW

This was due to a comparison between a long long value and an integer value inside the C parser. By changing the integer to a long long value, this was fixed.

# CLICKJACKING

been rectified through the updated security policy and the X FRAME-OPTIONS which have been set to DENY.

**THANKS FOR YOUR  
ATTENTION!**



# FIXES



**OTG-CONFIG-003**

# **FILE EXTENSIONS HANDLING FOR SENSITIVE INFORMATION**

## **Changed files**

---

N/A

N/A

This vulnerability was due to an error in setting up the github directory, this was remediated by editing the DocumentRoot value in the apache settings



**OTG-CONFIG-007**

## **HTTP STRICT TRANSPORT SECURITY**

### **Changed files**

---

N/A

N/A

This check will no longer be needed as the HTTP option is no longer an option on the website, as the website is only available in HTTPS.



**OTG-AUTHN-001**

# **CREDENTIALS TRANSPORTED OVER AN ENCRYPTED CHANNEL**

## **Changed files**

---

N/A

N/A

This has been fixed by the implementation of HTTPS on the website,  
and credentials are no longer sent in clear text.



# OTG-AUTHN-003

## WEAK LOCKOUT MECHANISM

### Changed files

/gnb/project/employee/manage_blocked.php	New file
/gnb/project/js/employee.js	+lines 69-84
/gnb/project/models/user.php	+lines 141-154
/gnb/project/db.php	+lines 287-366

A lockout mechanism was implemented: after failing a login 5 times in a row (status is kept in database), a user account automatically gets blocked. A new section in the employee area was added in order to allow employees to unblock user accounts which have been blocked.



OTG-AUTHN-004

## BYPASSING AUTHENTICATION SCHEMA

### Changed files

---

/gnb/project/login.php	+lines 19
/gnb/project/logout.php	+lines 5

The issue here was the constant SESSION\_ID. This has now been fixed, and a new SESSION ID is generated every time.



# OTG-AUTHN-005

## REMEMBER PASSWORD FUNCTIONALITY

### Changed files

---

N/A

N/A

This was also remediated through the implementation of HTTPS,  
see OTG-AUTHN-001.



**OTG-AUTHN-007**

# **WEAK PASSWORD POLICY**

## **Changed files**

---

/gnb/project/js/registration.js	+lines 21-45
---------------------------------	--------------

/gnb/project/registration/registration_request.php	+lines 3-15
--	-------------

In order to prevent weak passwords, the user is now forced to enter a password between 8 and 20 characters long, containing at least 1 number and 1 letter. Any special character is allowed. The password strength check is performed both on client side and server side during the registration process.



**OTG-AUTHZ-001**

# **DIRECTORY TRAVERSAL/FILE INCLUDE**

## **Changed files**

---

N/A

N/A

This was fixed by the same fix implemented in OTG-CONFIG-003.



# OTG-AUTHZ-002

## BYPASSING AUTHORIZATION SCHEMA

### Changed files

/gnb/project/employee:	+lines
client_details.php, client_transaction_details.php, employee_area.php, employee_overview.php, manage_clients.php, manage_registration.php, manage_transfer.php, transfer_details.php, search_client.php	5-21
/gnb/project/accounts:	+lines
account_overview.php, download_transactions.php, my_accounts.php, new_transaction.php, new_transaction_multiple.php, transaction_history.php, transaction_view.php, verify_transaction.php	5-16
/gnb/project/client/client_overview.php	+lines
	6-21
/gnb/project/registration:	+lines
registration_default.php, registration_pin.php, registration_request.php	5-14

Pages, sections and frames are no longer directly accessible, since checks are performed at the beginning of each php file: only users with the correct permissions are allowed to access specific pages.



**OTG-AUTHZ-004**

# **INSECURE DIRECT OBJECT REFERENCES**

## **Changed files**

---

N/A

N/A

Please refer to OTG-CRYPST-003 for this fix, since the vulnerability was based on the usage of an unencrypted communication channel.



**OTG-SESS-001**

# **BYPASSING SESSION MANAGEMENT SCHEMA**

## **Changed files**

---

N/A

N/A

This was also remediated through the implementation of HTTPS,  
see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.



# OTG-SESS-002

## COOKIES ATTRIBUTES

### Changed files

---

N/A

N/A

This was also remediated through the implementation of HTTPS,  
see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.



# OTG-SESS-003

## SESSION FIXATION

### Changed files

---

/gnb/project/login.php	+lines 19
/gnb/project/logout.php	+lines 5

This was fixed by the same fix in OTG-AUTHN-004.  
SESSION ID is now renewed each time.



# OTG-SESS-004

## EXPOSED SESSION VARIABLES

### Changed files

---

N/A

N/A

This was also remediated through the implementation of HTTPS,  
see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.



## OTG-SESS-005

# CROSS SITE REQUEST FORGERY

### Changed files

---

/gnb/project/accounts/new_transaction.php	+lines 17-24, 54
/gnb/project/accounts/new_transaction_multiple.php	+lines 21-27, 66-73, 130
/gnb/project/accounts/verify_transaction.php	+lines 16-21, 59, 108, 146, 165

This issue has been resolved by adding a randomly generated token to all the forms. This token is checked on receiving the values of the form to detect Cross Site Request Forgery.



# OTG-SESS-007

## SESSION TIMEOUT

### Changed files

---

config/php5/apache2/php.ini

line 1515

This has been fixed by modifying the PHP settings file.



# OTG-INPVAL-001

## REFLECTED CROSS SITE SCRIPTING

### Changed files

---

/gnb/project/genericfunctions.php

new file (+lines 10-103)

Reflected XSS worked on the verify\_transaction.php page by printing the description inserted by the user on the previous page. By performing input sanitization on every request parameter, this attack is not longer possible.



# OTG-INPVAL-002

## STORED CROSS SITE SCRIPTING

### Changed files

---

/gnb/project/genericfunctions.php new file (+lines 10-103)

Similarly to the reflected XSS, the site was vulnerable because of lack of input sanitization. All user inputs get now properly sanitized, removing html characters.



# OTG-INPVAL-005

## SQL INJECTION

### Changed files

/gnb/project/db.php	new file (+lines 1-1092)
/gnb/config/database/setup.php, /gnb/project/accounts/account_overview.php, /gnb/project/accounts/download_transactions.php, /gnb/project/accounts/my_accounts.php, /gnb/project/accounts/new_transaction.php, /gnb/project/accounts/transaction_history.php, /gnb/project/accounts/transaction_view.php, /gnb/project/accounts/verify_transaction.php, /gnb/project/authentication.php, /gnb/project/awesome_data.php, /gnb/project/employee/client_details.php, /gnb/project/employee/client_transaction_details.php, /gnb/project/employee/manage_registration.php, /gnb/project/employee/manage_transfer.php, /gnb/project/employee/search_client.php, /gnb/project/employee/transfer_details.php, /gnb/project/models/account.php, /gnb/project/models/transaction.php, /gnb/project/models/user.php, /gnb/project/registration/registration_request.php, /gnb/project/resource_mappings.php	modified all lines with "DB::i()->"
/gnb/project/dbheader.php	removed file

This issue has been resolved by using prepared statements instead of pure `mysql_*` commands. This was accomplished by using PDO and its functionality for building prepared statements. This validates the input for its data type and prevents SQL injections.



# OTG-INPVAL-013

## COMMAND INJECTION

### Changed files

---

/gnb/project/accounts/new_transaction_multiple.php	-lines 20-27 +lines 21-30
--	------------------------------

Since command injection was possible by injecting commands into filenames, this is no longer possible as each uploaded file gets automatically renamed. Even though the c parser receives additional parameters, these are passed directly by the application and cannot be injected by an attacker.



# OTG-INPVAL-014

## BUFFER OVERFLOW

### Changed files

---

gnb/project/lib/ctransact/src/ctransact.c

+line 284

The application was never vulnerable to buffer overflow, but only to integer overflow. This was due to a comparison between a long long value and an integer value inside the c parser. By changing the integer to a long long value, this is no longer possible.



# OTG-INPVAL-015

## INCUBATED VULNERABILITIES

### Changed files

---

See fix slides OTG-INPVAL 001, 002, 005

By having fixed the SQL injection and stored XSS vulnerabilities, this vulnerability was automatically fixed. Please refer to the fix slides concerning OTG-INPVAL 001, 002, 005.



**OTG-CRYPST-003**

# **SENSITIVE INFORMATION SENT VIA UNENCRYPTED CHANNELS**

## **Changed files**

---

N/A

N/A

This was also remediated through the implementation of HTTPS, see OTG-AUTHN-001. SESSION ID is no longer sent in clear text.



# OTG-BUSLOGIC-001

## BUSINESS LOGIC DATA VALIDATION

### Changed files

---

/gnb/project/dp.php +lines 1234-1237

In order to prevent a user incrementing his balance by sending money to his own account, we added a check before performing a transaction. A user is not allowed to use his own account as a destination anymore.



# OTG-BUSLOGIC-002

## ABILITY TO FORGE REQUESTS

### Changed files

---

N/A

N/A

This was fixed with the same fix as OTG-SESS-005, see section for more details.



# OTG-BUSLOGIC-004

## PROCESS TIMING

### Changed files

---

/gnb/project/db.php +line 224,238-246

This was fixed by adding a random waiting time to all login requests.



# OTG-BUSLOGIC-007

## DEFENSES AGAINST APPLICATION MIS-USE

### Changed files

---

N/A

N/A

Since the described vulnerability was the same as the one presented in OTG-BUSLOGIC-001, please refer to the fix made in that section.



# OTG-BUSLOGIC-009

## UPLOAD OF MALICIOUS FILES

### Changed files

---

/gnb/project/accounts/new_transaction_multiple.php	-lines 20-27
	+lines 21-30

Each uploaded file now gets automatically renamed. Furthermore, after a batch transaction process finished, the uploaded file gets removed from the uploads folder.



# OTG-CLIENT-009

## CLICKJACKING

### Changed files

---

/gnb/config/apache2/httpd.conf

+lines 1-7

This has been rectified through the updated security policy and The X-  
FRAME-OPTIONS has been set to DENY.



# USE-CASES



# PASSWORD RECOVERY

Goal	Reset a users password in case it is not remembered any longer.
Actors	All users (Customers and Employees)
Pre-Conditions	User account must exist and must know his PIN.
Main course of execution	On the login page click on "Forgot your brocode?" and enter your email address. You will receive instructions to reset your password by mail.
Alternate course	n/a
Post conditions	Password is set to a new password by the user.
Exceptions	Clicking on the received reset hash took longer than 1 day. It is necessary to repeat the whole procedure.
Data format used	n/a



# ENCRYPTED TAN GENERATION AND DELIVERY VIA E-MAIL TO CUSTOMER

Goal	Send the TANs in an encrypted PDF to the customer.
Actors	Customers that are using standard TANs.
Pre- Conditions	Customer is approved and did select "TANs" as banking method at the registration.
Main course of execution	Create new account, wait for approval by employee, log in and copy the shown password in order to open the encrypted PDF that was received by mail.
Alternate course	n/a
Post conditions	The customer has access to his TANs provided by the decrypted PDF.
Exceptions	n/a
Data format used	Encrypted PDF file



# DOWNLOAD OF SCS AFTER REGISTRATION

Goal	Download the SmartCardSimulator after SCS was selected as the banking method.
Actors	Customers that are using the SCS.
Pre-Conditions	Customer is approved and did select "SCS" as banking method at the registration.
Main course of execution	Create new account, wait for approval by employee and log in. Write down the provided PIN and download the provided Java-Application "SmartCardSimulator".
Alternate course	n/a
Post conditions	The customer has access to the SCS and can generate TANs using his PIN, the destination account number and the transaction amount OR a transaction batch file including the aforementioned information.
Exceptions	n/a
Data format	Java Application (requires Java 1.7)



# TRANSFER USING SCS (SINGLE TRANSACTION)

Goal	Create transactions using the SmartCardSimulator.
Actors	Customers that are using the SCS.
Pre-Conditions	Customer is approved, did select "SCS" as banking method at the registration, downloaded the SCS and remembers the PIN.
Main course of execution	Customers enter the transaction details (destination account, transaction amount) and their PIN into the web interface and the SCS, and copy the TAN generated by the SCS into the web interface. The transaction is then posted and executed.
Alternate course	see "Transfer using SCS - Batch transaction"
Post conditions	The customer did create transactions using the SmartCardSimulator.
Exceptions	n/a
Data format used	Java Application (requires Java 1.7), batch processing format see "Batch transfers should allow multiple transfers via the same uploaded file"



# TRANSFER USING SCS (BATCH TRANSACTION)

Goal	Execute multiple transactions sequentially by uploading a batch transaction file. The legitimacy of these transactions is verified by a TAN generated by the SCS.
Actors	Customer
Pre-Conditions	Customer must be logged in, have sufficient funds for all transactions and remember his PIN displayed at registration.
Main course of execution	Logged in Customer switches to "My Accounts" page, selects an account, selects "New transaction (multiple)", chooses a file. He also enters a tan, which is generated by additionally opening the batch file in the SCS application in combination with his pin.
Alternate course	Single transaction method. Entering every transaction individually via the web form as described above
Post conditions	None
Exceptions	Invalid batch file format. Insufficient funds.
Data sources	Data format explained in detail on the next slide and on the "New transaction (multiple)" page.



# BATCH TRANSACTION: DATA FORMAT

- Each transaction in a separate line
- Fields separated only by a comma
- No quoting of whitespace required
- The complete input between two commas gets treated as value
- Example: DST\_ACCOUNT,AMOUNT,DESCRIPTION



# TIME-TRACKING



# ALEXANDER LILL

Task	Time	
Research fix: SQL Injection	1h	Fix: Lockout - Add unblock user functionality
Fix: SQL Injection - Login / Misc	1h	1h
Fix: SQL Injection - User functions	1h	Fix: Lockout functionality
Fix: SQL Injection - Account functions	1h	1h
Fix: SQL Injection - TAN functions	1h	Testing Lockout and unblock functionality
Fix: SQL Injection - Transaction functions	1h	Feature: SCS TAN handling (DB)
Fix: SQL Injection - Overview functions	0,5h	1h
Use DB as Singleton object and change all calls	1h	Feature: Implement reset password functionality (DB)
Adjust DB schema to new requirements	1h	1h
Add PIN functionality to DB and PHP	1h	Prepare presentation
		1h
		Prepare deliverables
		1h
		Fixing balanice initialization bugs
		1h
		Adding empty templates for vulnerabilitis to presentation
		1h
		Fix: Process timing
		1h
		<b>TOTAL</b>
		<b>19,5h</b>



# LORENZO DONINI

Task	Time		
Group meeting	1h	Java SCS: GUI	1h
Fix: resource mappings	1h	Java SCS: Business logic	1h
Fix: session	0.5h	Java SCS: TAN generation	1h
Fix: password	0.5h	Java SCS: Testing	1h
Fix: lockout mechanism	1h	Transactions: single transactions logic changed	1h
Features/Use-cases testing	1h	Transactions: batch transactions logic changed	1h
Features/Use-cases bugfixing	1h	Initialization of client balance added	1h
Registration: new business logic	1h	Presentation: core slides	1h
Registration: new GUI	1h	Presentation: fix slides	1h
Registration: testing & fixing	1h	TOTAL	18h



# FLORIAN MAURACHER

Task	Time	
Group meeting	1h	Fix vulnerabilities in fileupload 1h
Read BBTesting report Team5	0.5h	Multiple transactions, one TAN (PHP) 1h
Discuss Phase3 and coordinate work	1h	Multiple transactions, one TAN (C) 1h
Fix mail delivery	1h	Verify SCS TANs for batch transactions 1h
Secure apache2 configuration	1h	Debug SCS logic for batch transactions 1h
Secure php configuration	1h	Fix timestamps for batch transactions 0.5h
Prepare VM for Phase3	1h	Presentation: Fixes 1h
Develop mechanism for TAN generation	1h	Presentation: Review 1h
Discuss mechanism for TAN generation	1h	Prepare delivery VM 1h
		<b>TOTAL</b> 17h



# MAHMOUD NASER

[F]ix, [R]esearch, [I]mplement	
[T]est, [P]resentation, [D]esign	
[R] PW PDF	1h
[I] FPDI library	1h
[R] bugs with FPDI	1h
[I] FPDI bug fixes	1h
[I] FPDF library	1h
[T] FPDF PW protection	1h
[T] PDF PW on GNB Website	1h
[F] PDF PW bugs	1h
[D] PW reset logic	0.5h
[D] PW reset pages	1h
[D] PW forgot logic	0.5h
[D] PW forgot pages	1h
[D] email template	1h
[C] Token generetaion mechanism	1h
[F] PW email bugs	1h
[C] Transaction history fields	1h
[C] PDF transaction history fields	1h
[F] Transaction bugs	1h
[P] adding content	1h
<b>TOTAL</b>	<b>18h</b>

