SHALL WE PLAY A GAME?

whoami

- Tamas Szakaly (sghctoma)
- from Hungary, the land of Pipacs and gulash, and ….
- OSCE
- pentester/developer @ PRAUDIT
- part of team Prauditors, European champion of Global Cyberlympics 2012

whoami

- a binary guy
- loves crackmes and copy protections
- “I am not a computer nerd. I prefer to be called a hacker!”
-

whoami

- a binary guy
- loves crackmes and copy protections
- “I am not a computer nerd. I prefer to be called a hacker!”
-

prepare for big coming out:

whoami

- a binary guy
- loves crackmes and copy protections
- “I am not a computer nerd. I prefer to be called a hacker!”
-

prepare for big coming out:

I’ve been in love with the Win32 API for years

game modding

- the urge to make things better
- implement your own ideas
- custom content: maps, models, etc.

to create

game modding

Create, discover, and download content for your game

- the urge to make things better
- implement your own ideas
- custom content: maps, models, etc.

to create

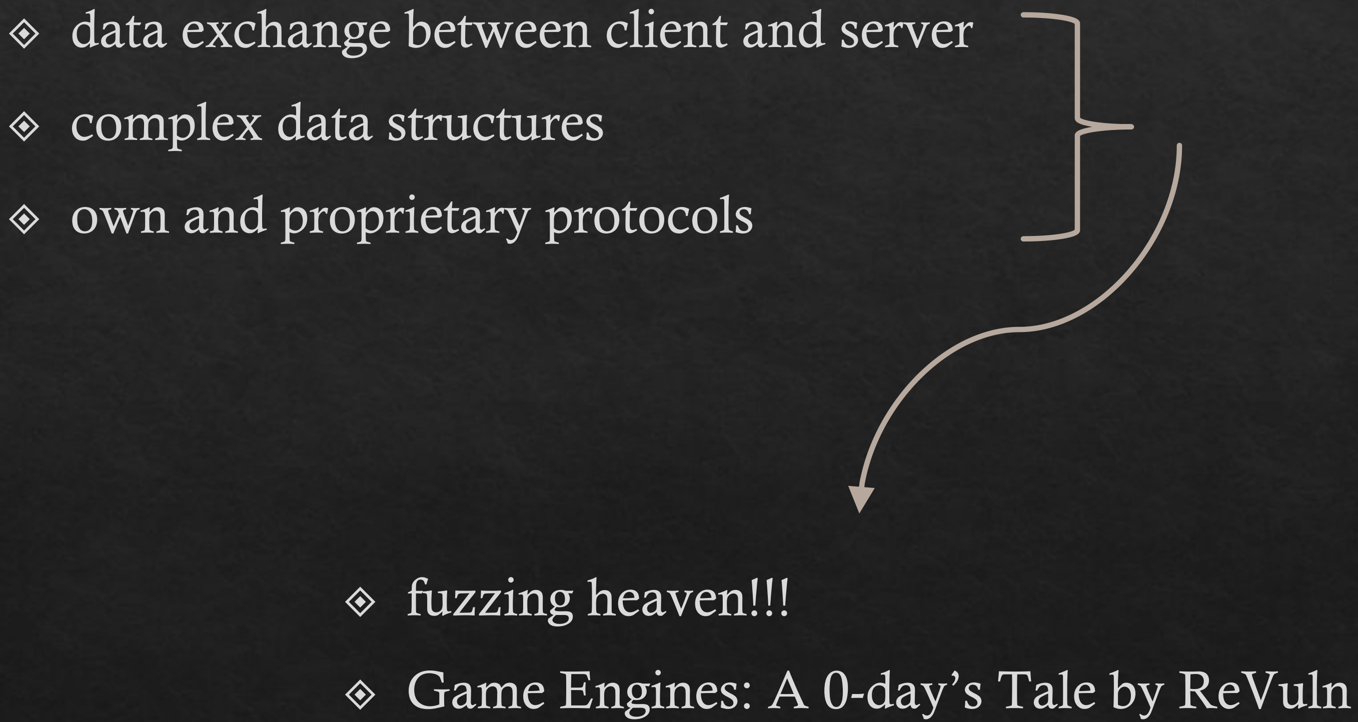
- share with others
  - http://www.moddb.com/
  - http://www.gamemodding.net/
- even get paid for them
  - Steam Workshop

to share

nobody plays alone

- data exchange between client and server
- complex data structures
- own and proprietary protocols

nobody plays alone



- data exchange between client and server
- complex data structures
- own and proprietary protocols

- fuzzing heaven!!!
- Game Engines: A 0-day’s Tale by ReVuln

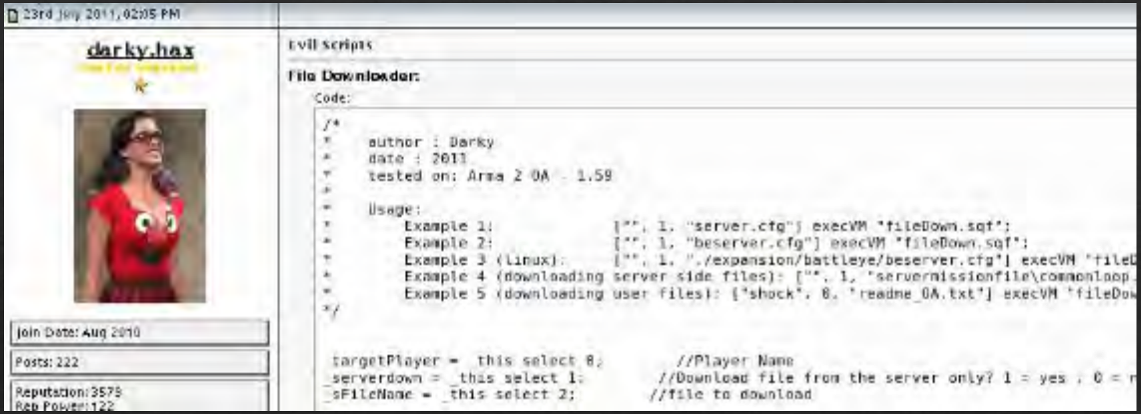
scripting in games

- built-in scripting engines
- custom-made or embedded language
  - ARMA scripts
  - Lua-scripted video games @Wikipedia - 153 titles
  - Squirrel (Valve games)
- purpose: dynamic maps, AI, etc.
- available to modders

could scripts be really dangerous?

- downloaded from the server, or with custom maps
- runs on the gamer’s machine
- dangerous functionality (e.g. file I/O)
- poorly implemented sandboxes
- easy to exploit: no need to circumvent exploit mitigations

surely I’m not the first one …



23rd July 2011, 02:05 PM

darky.hax

Join Date: Aug 2010

Posts: 222

Reputation: 3579
Rep Power: 122

Evil scripts

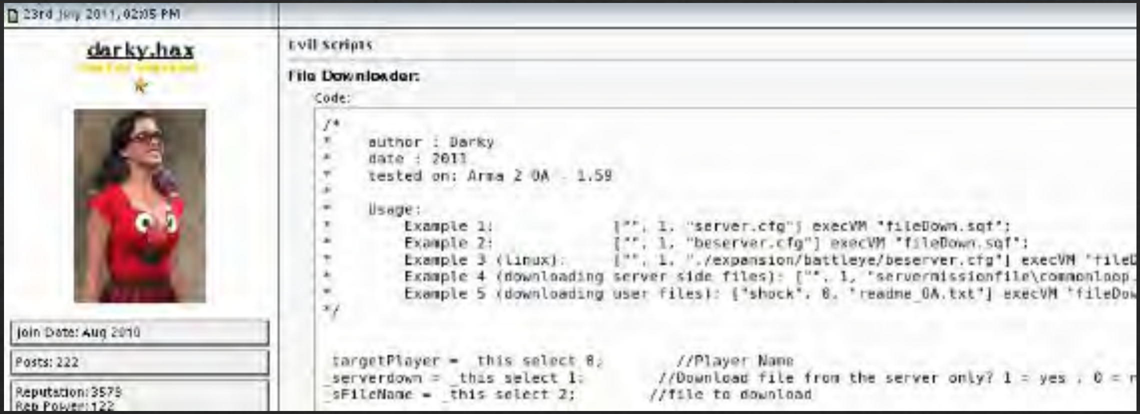
File Downloader:

Code:

```
/*
*     author : Darky
*     date : 2011
*     tested on: Arma 2 OA - 1.59
*
*     Usage:
*         Example 1:              ["", 1, "server.cfg"] execVM "fileDown.sqf";
*         Example 2:              ["", 1, "beserver.cfg"] execVM "fileDown.sqf";
*         Example 3 (linux):      ["", 1, "./expansion/battleye/beserver.cfg"] execVM "fileD
*         Example 4 (downloading server side files): ["", 1, "servermissionfile\commonloop.
*         Example 5 (downloading user files): ["shock", 0, "readme_OA.txt"] execVM "fileDow
*/


 _targetPlayer = _this select 0;        //Player Name
_serverdown = _this select 1;        //Download file from the server only? 1 = yes , 0 = n
_sFileName = _this select 2;        //file to download
```

surely I'm not the first one …



23rd July 2011, 02:05 PM

darky.hax

Join Date: Aug 2010

Posts: 222

Reputation: 3579
Rep Power: 122

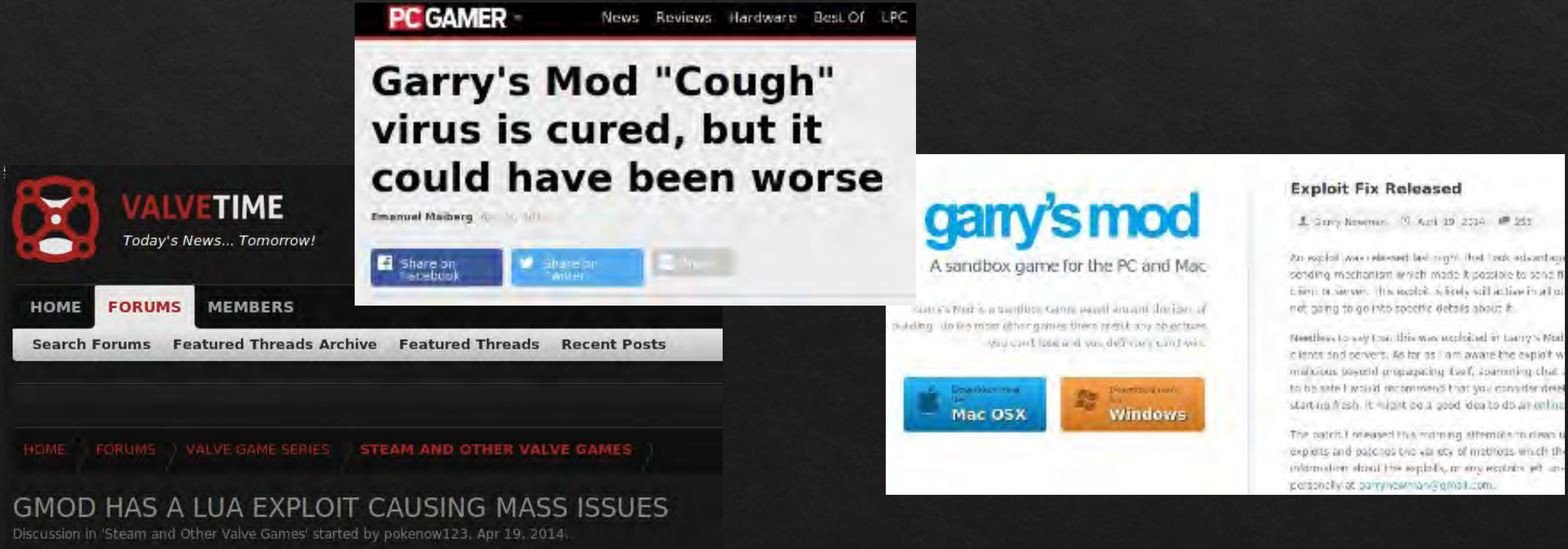
Evil scripts

File Downloader:

Code:

```
/*
*    author : Darky
*    date : 2011
*    tested on: Arma 2 OA - 1.59
*
*    Usage:
*        Example 1:                   ["", 1, "server.cfg"] execVM "fileDown.sqf";
*        Example 2:                   ["", 1, "beserver.cfg"] execVM "fileDown.sqf";
*        Example 3 (linux):           ["", 1, "./expansion/battleye/beserver.cfg"] execVM "fileD
*        Example 4 (downloading server side files): ["", 1, "servermissionfile\commonloop.
*        Example 5 (downloading user files): ["shock", 0, "readme_OA.txt"] execVM "fileDow
*/


_targetPlayer = _this select 0;         //Player Name
_serverdown = _this select 1;         //Download file from the server only? 1 = yes ; 0 = n
_sFileName = _this select 2;        //file to download
```



PC GAMER
News Reviews Hardware Best Of LPC

Garry's Mod "Cough" virus is cured, but it could have been worse

Emanuel Maiberg

Share on Facebook
Share on Twitter

VALVETIME
Today's News... Tomorrow!
HOME FORUMS MEMBERS
Search Forums Featured Threads Archive Featured Threads Recent Posts
HOME › FORUMS › VALVE GAME SERIES › STEAM AND OTHER VALVE GAMES ›

GMOD HAS A LUA EXPLOIT CAUSING MASS ISSUES

Discussion in 'Steam and Other Valve Games' started by pokenow123, Apr 19, 2014.

garry's mod
A sandbox game for the PC and Mac
Download now for Mac OSX
Download now for Windows

Garry's Mod is a sandbox Game based around the idea of building. Unlike most other games there aren't any objectives. You can't lose and you definitely can't win.

Exploit Fix Released

Garry Newman · April 19, 2014 · 255

An exploit was released last night that took advantage
sending mechanism which made it possible to send fi
client to server. This exploit is likely still active in all u
not going to go into specific details about it.

Needless to say that this was exploited in Garry's Mod
clients and servers. As far as I am aware the exploit w
malicious beyond propagating itself, spamming chat
to be safe I would recommend that you consider dele
starting fresh. It might be a good idea to do an online

The patch I released this morning attempts to clean u
exploits and patches the variety of methods which th
information about the exploits, or any exploits, let
personally at garrynewman@gmail.com.

… so, why do this talk?

- game exploits are used to cheat

... so, why do this talk?

- game exploits are used to cheat
- but they can give access to your pc

... so, why do this talk?

- game exploits are used to cheat
- but they can give access to your pc
- also a gateway to your home network
  - other computers
  - routers
  - phones (VOIP and mobile)
  - TV sets
  - smart house components
  - security cameras

… so, why do this talk?

- game exploits are used to cheat
- but they can give access to your pc
- also a gateway to your home network
  - other computers
  - routers
  - phones (VOIP and mobile)
  - TV sets
  - smart house components
  - security cameras

nobody seems to talk about this!!!

no sandbox in Sandbox

- target: Crysis 2 and the whole CryEngine3
- uses Lua as a scripting engine
- no sandbox whatsoever
- yes, we can even call os.execute

attack scenario

- gamer plays a custom-made Crysis 2 mod
- a push of a button triggers some Lua code
- the Lua code starts a netcat connectback shell

PRESS START >

PWNED!
d with
CRYENGINE® 3

one of the reasons I love Win32

- Win32 APIs that work with files accept UNC paths
- yes, LoadLibrary and ShellExecute do too
- no need to write shellcode, we can load a DLL from a remote share
- or execute something from a remote share
- side effect: we can steal NT hashes

slide #23

disclaimer #1: intentionally left (almost) blank, didn’t want to fly in the face of fate.

disclaimer #2: no, I do not believe in the 23 Enigma, this slide is an attempted joke.

disclaimer #3: yes, I do realize that this intentionally-left-blank slide has more content than most of the others.

PRESS START

the kobold who hijacked DLLs

- target: DOTA2
- another Lua-scriptable game
- there is a sandbox, but its leaky
- we can use the standard io library
    - use the SMB NT hash stealing trick
    - steal files
    - deploy autorun stuff
    - etc…

attack scenario

- malicious game mode with some Lua scripts
- a Base64-encoded PE file is decoded
- and the game's main exe is overwritten with it
- so the next time the game starts, the game does not start
- instead the Mighty Calculator is unleashed on the gamer

PRESS START >

from crash to exploit

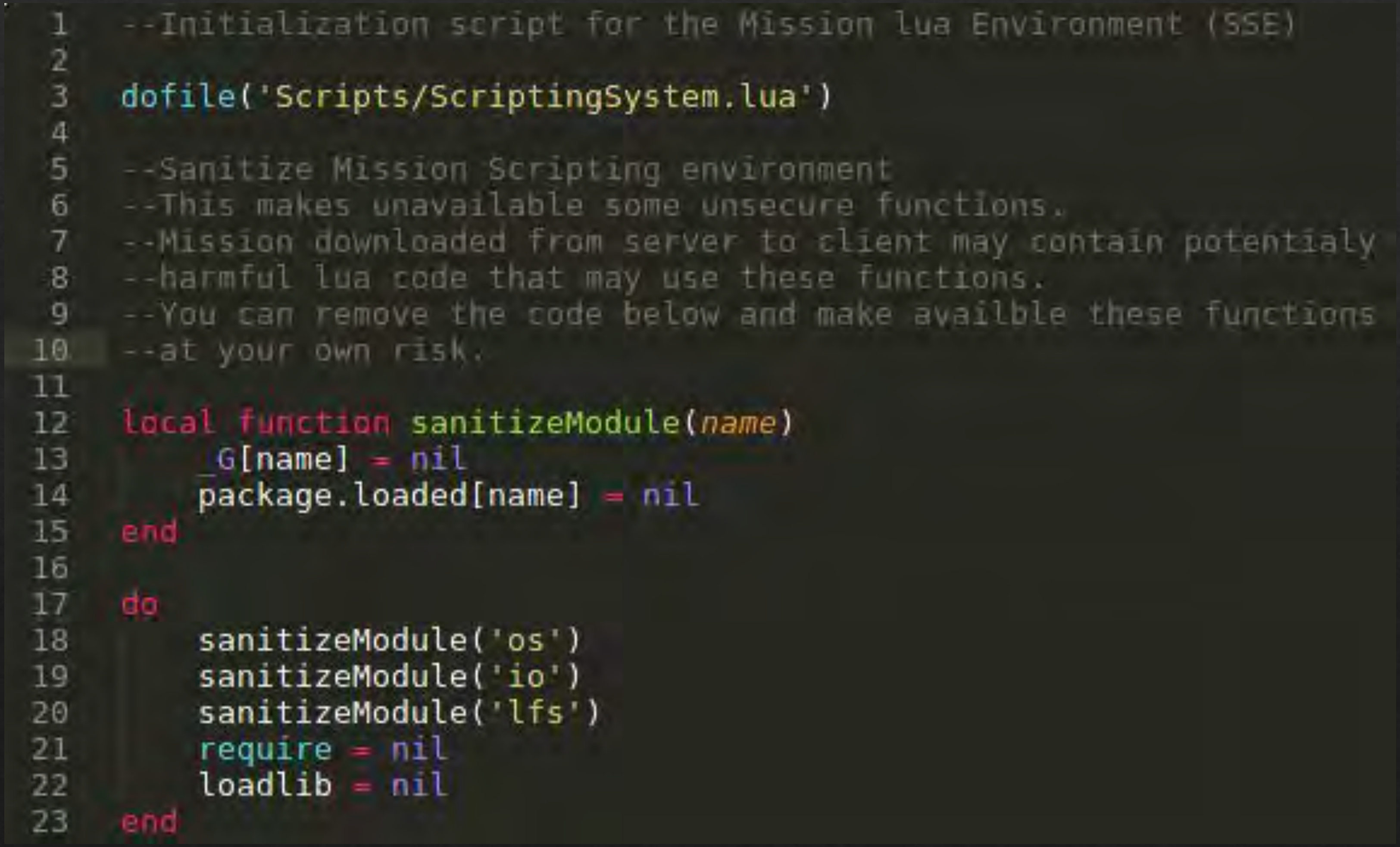
- target: Digital Combat Simulator (DCS World)
- THE combat flight simulator
- uses Lua for mission scripting
- another leaky sandbox
- reported one issue, found another one

attack scenario

- gamer joins a server that serves a malicious map
- a Lua script is attached to the “plane crash” event
- plane crashes, Lua executes code on gamer’s machine
- the Industry Standard Exploit Testing Tool* launches

* it’s called Calculator by the uninitiated

quiz: where is the leak?



```
--Initialization script for the Mission lua Environment (SSE)

dofile('Scripts/ScriptingSystem.lua')

--Sanitize Mission Scripting environment
--This makes unavailable some unsecure functions.
--Mission downloaded from server to client may contain potentialy
--harmful lua code that may use these functions.
--You can remove the code below and make availble these functions
--at your own risk.

local function sanitizeModule(name)
    _G[name] = nil
    package.loaded[name] = nil
end

do
    sanitizeModule('os')
    sanitizeModule('io')
    sanitizeModule('lfs')
    require = nil
    loadlib = nil
end
```

quiz - backup question #1

The title of this talk is a quote - who asked that question?

quiz - backup question #2

what is my favorite movie?

PRESS START >

when the gamer is the bad guy

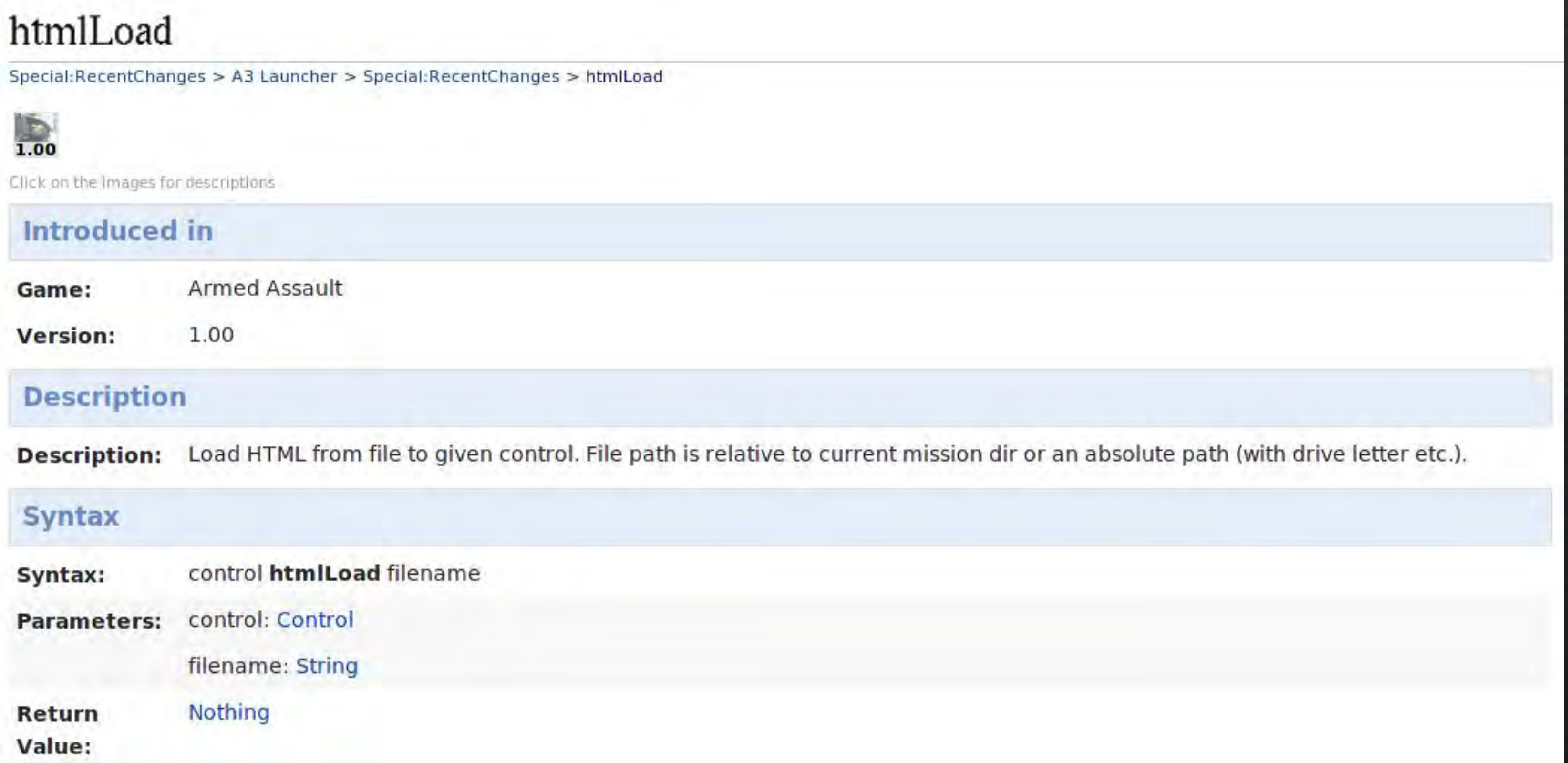
- target: Armed Assault 3 (ARMA3)
- military combat simulator
- customizable squads (name, URL, logo, etc.)
- squad info from user-supplied URL
- squad info is XML.. so, XXE? nope :(
- but hey, it’s an SSRF :)

attack scenario

- based on real-life experiences
- ARMA3 server + local-only PHP-Charts
- RCE via GET request in PHP-Charts
- give exploit-triggering URL as squad Info URL
- join the server, and profit!

PRESS START >

more, I want moooooore!



htmlLoad

Special:RecentChanges > A3 Launcher > Special:RecentChanges > htmlLoad

1.00

Click on the images for descriptions

Introduced in

Game: Armed Assault

Version: 1.00

Description

Description: Load HTML from file to given control. File path is relative to current mission dir or an absolute path (with drive letter etc.).

Syntax

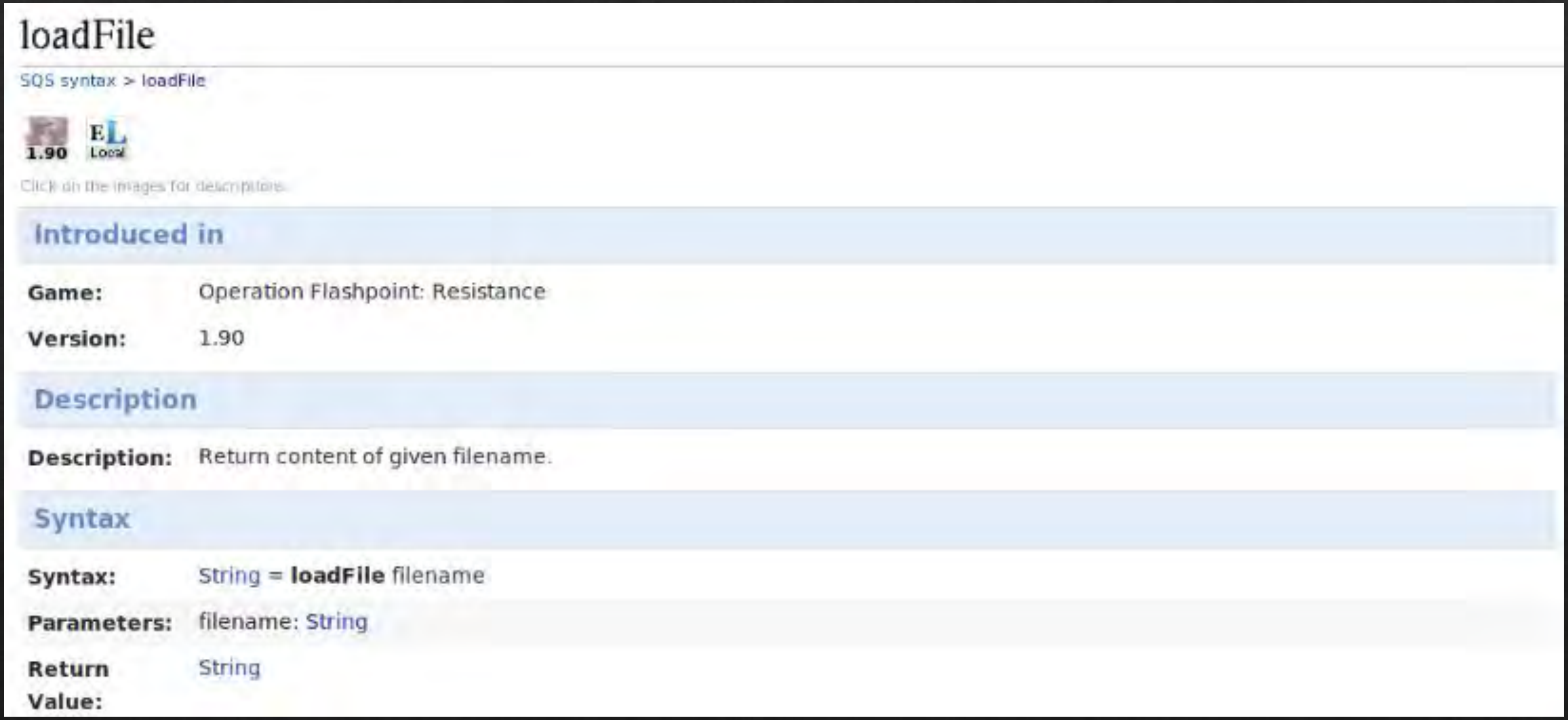
Syntax: control **htmlLoad** filename

Parameters: control: Control

filename: String

Return Value: Nothing

more, I want moooooore!



loadFile

SQS syntax > loadFile

Click on the images for descriptions.

Introduced in

Game:	Operation Flashpoint: Resistance
Version:	1.90

Description

Description:	Return content of given filename.

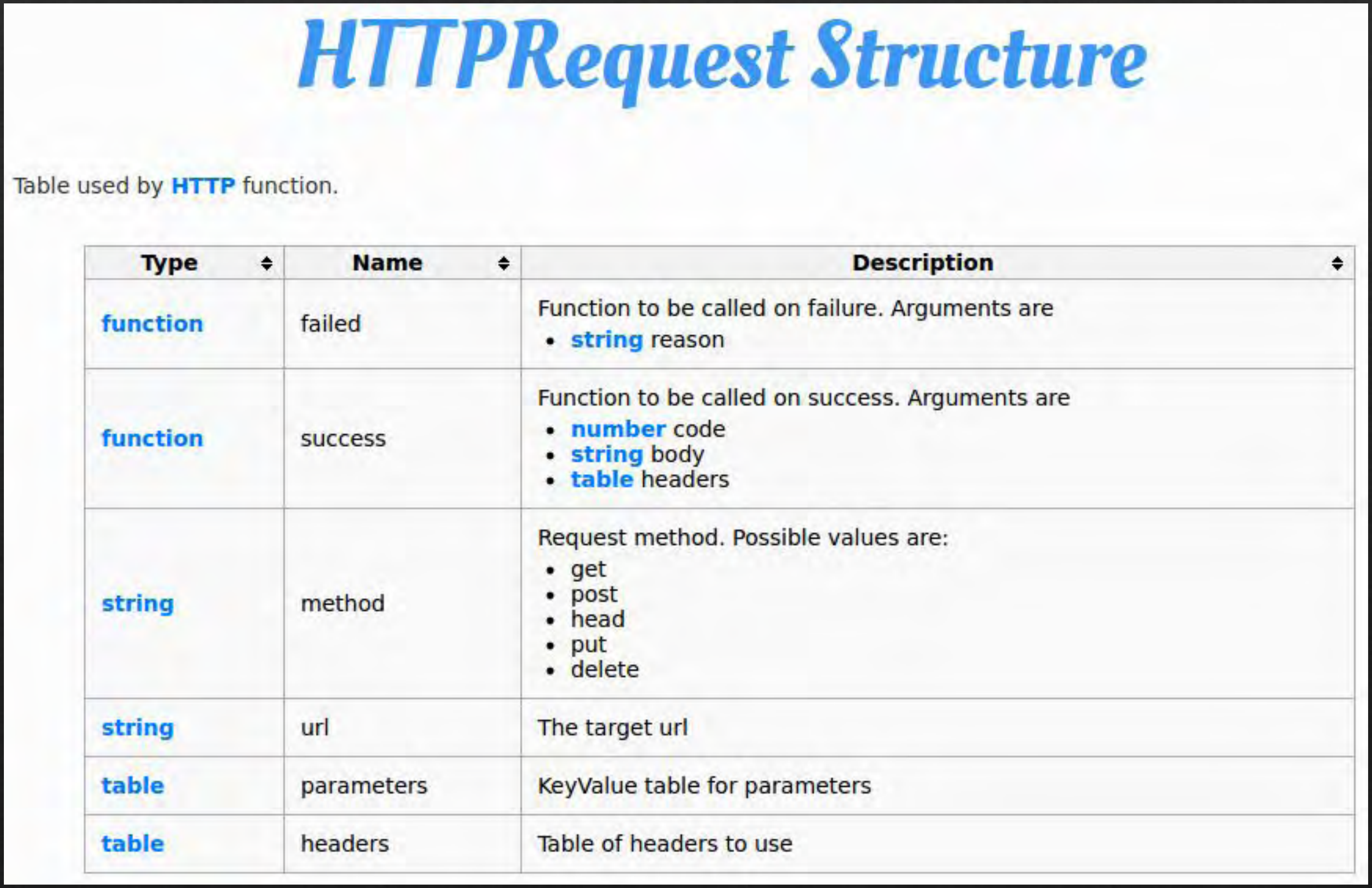
Syntax

Syntax:	String = **loadFile** filename
Parameters:	filename: String
Return Value:	String

spy game

- target: Garry’s Mod
- a sandbox game based on Source Engine
- lots of Lua-related bugs
- lots of mitigations:
  - custom implementation for dangerous function (e.g. package.loadlib)
  - restricted file I/O (directory traversal was possible, now it isn't)
  - proper Lua sandbox

tight sandbox, what to abuse?



HTTPRequest Structure

Table used by **HTTP** function.

Type	Name	Description
function	failed	Function to be called on failure. Arguments are • **string** reason
function	success	Function to be called on success. Arguments are • **number** code • **string** body • **table** headers
string	method	Request method. Possible values are: • get • post • head • put • delete
string	url	The target url
table	parameters	KeyValue table for parameters
table	headers	Table of headers to use

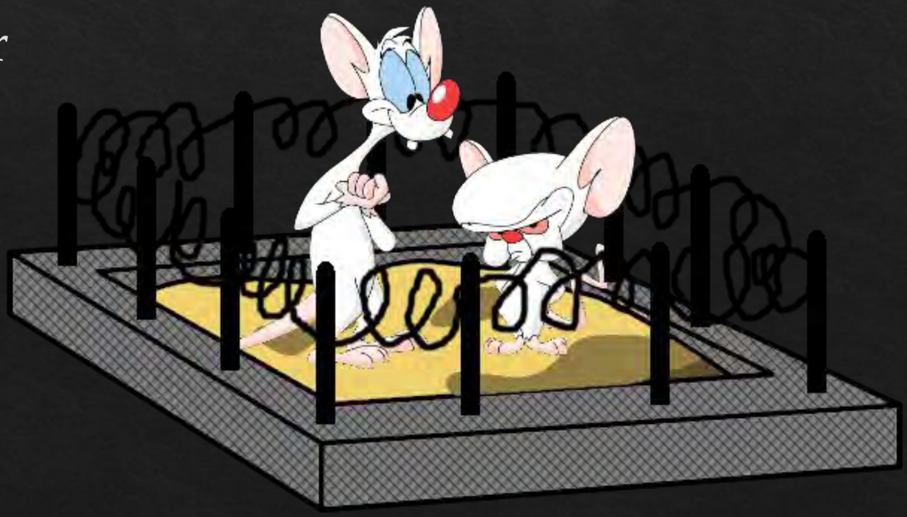
attack scenario

- evil GMod server admin
- the game becomes an HTTP proxy to the player’s network
- scans every connecting player’s network
- brute forces HTTP basic authentication
- steals images from security cameras

PRESS START

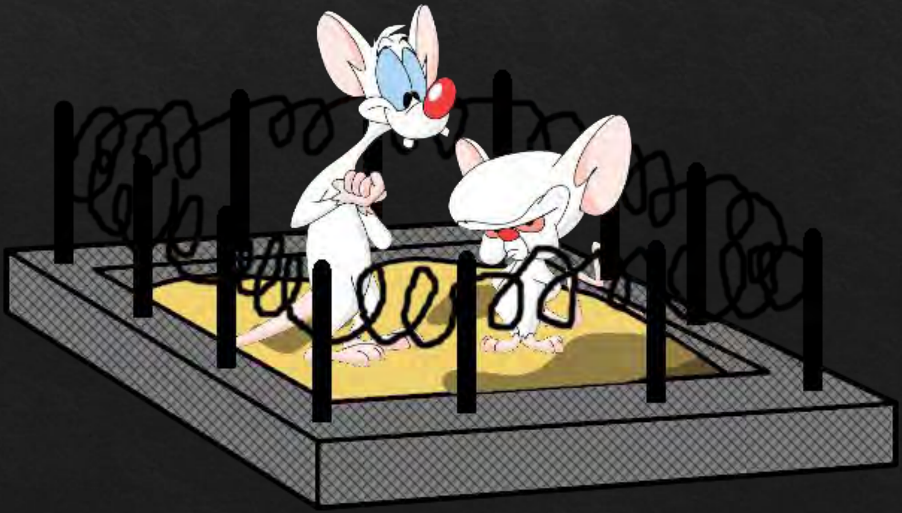
you should be afraid of mice

- target: Logitech Gaming Software
- not a game, but a gaming mouse
- can create profiles for all G-series Logitech peripherals
- a Lua script is attached to these profiles
- can script peripheral behavior
- very tight Lua sandbox



attack scenario

- gamer downloads a malicious profile
- activates it
- a certain button press triggers the exploit
- and again, the Industry Standard Exploit Testing Tool launches



@corsix's black magic

- a beautiful Lua sandbox escape by **@corsix** (CoH2 exploit)
- he abused handcrafted Lua bytecode
  1. string.dump to get bytecode string
  2. modify bytecode
  3. loadstring to load modified bytecode

@corsix's black magic

- get memory address of variable as double
- hand-craft arbitrary UpVals

@corsix's black magic

- get memory address of variable as double
- hand-craft arbitrary UpVals

arbitrary memory read-write

getting memory addresses

- in Lua, everything is a TValue
- bits 0-63: actual value (pointer to struct or a double)
- bits 64-95: type
- for loop: OP_FORPREP followed by OP_FORLOOP
- OP_FORPREP checks if parameters are numbers
- OP_FORLOOP treats parameters as numbers
- we can nop out OP_FORPREP by modifying bytecode!!
  - so everything gets treated as a number

crafting arbitrary TValues

- create a string that looks like an UpVal
  - the UpVal's TValue* will be the address we want to access
- get the address of the actual char array of that string
- create another string out of this address
  - after bytecode modification this will be interpreted as an LClosure
  - summary: **we have an UpVal that represents a TValue that points to an arbitrary memory location**

what did @corsix do?

- created a coroutine variable
    - creating a coroutine creates a CClosure
    - a CClosure represents a function pointer (luaB_auxwrap in this case)
- replaced the CClosure's function pointer with ll_loadlib
    - it is basically a LoadLibrary wrapper
- called the coroutine

what did I do differently?

- mine is a 64 bit exploit
    - memory layout (struct packing)
    - calling conventions
    - sizeof(double) = sizeof(void *) on 64bit
    - the latter makes the exploit much simpler on 64bit
- calling LoadLibrary directly instead of ll_loadlib

ll_loadlib vs LoadLibrary

- ANSI-only Lua: ll_loadlib is just a stub – can’t use it
- call native functions directly
  - prototype must match CClosure’s function pointer’s
    - one parameter, a pointer to the actual Lua state
  - LoadLibrary is a good candidate (has one pointer parameter)

calling LoadLibrary

- get LoadLibraryA’s address
- replace luaB_auxwrap with LoadLibraryA
- overwrite the Lua state with the DLL name
- call the coroutine

difficulties

- how to get LoadLibrary's address?
- how to get the address of the Lua state struct?
  - coroutine.running to the rescue
- seemingly random crashes
  - debug hooks have to be disabled
- more crashes
  - garbage collector has to be stopped
  - the overwritten Lua state has to be restored

getting LoadLibrary's address

- simple solution
  1. get address diff of LoadLibrary and luaB_auxwrap from PE
  2. read address of luaB_auxwrap at runtime
  3. the rest is elementary school math
- more generic solution (used in my Redis exploit)
  1. get address to NT header
  2. get address of Import Directory
  3. search for KERNEL32.DLL
  4. get LoadLibrary's address from IAT

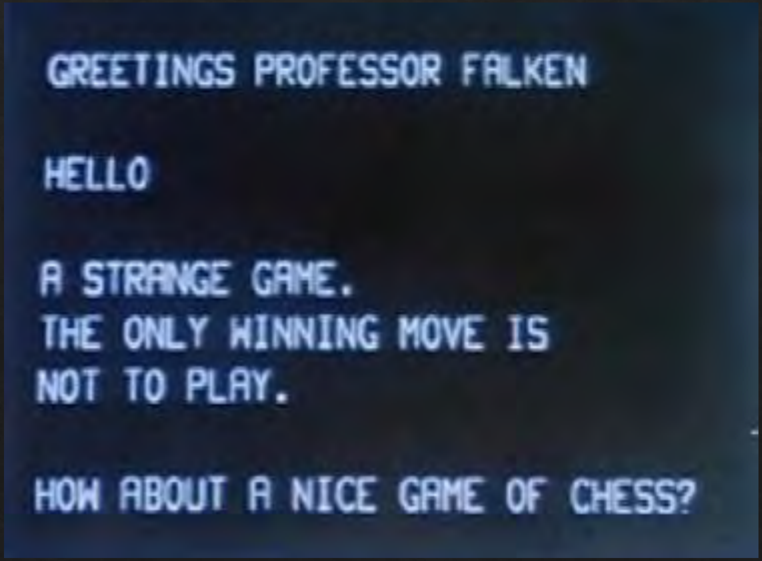
restrictions

- only 16 bytes of the Lua state can be overwritten
- so DLL path must be .le 15 (+1 null byte)
  - if we use LoadLibraryA instead of LoadLibraryW
- while using UNC paths
  - we can omit the .dll extension
  - e.g. \\evilhaxor\a\b
  - so we've got 9 characters for an IP, a NETBIOS. or a domain name

PRESS START

endgame

- should we listen to Joshua?
- sad truth: we should be security-conscious even while leisuring
  - don’t download anything from the Internet (duh!)
  - don’t play on untrusted servers
  - updates!! (Steam does this right)
- game devs: you should think through cool new features from a security standpoint too!



```
GREETINGS PROFESSOR FALKEN

HELLO

A STRANGE GAME.
THE ONLY WINNING MOVE IS
NOT TO PLAY.

HOW ABOUT A NICE GAME OF CHESS?
```

contact

- name: Tamas Szakaly
- mail: tamas.szakaly@praudit.hu
- mail: sghctoma@gmail.com
- PGP fingerprint:
  4E1F 5E17 7A73 2C29 229A CD0B 4F2D 6CD0 9039 2984
- twitter: @sghctoma

links & credits

- http://www.moddb.com/
- http://www.gamemodding.net/
- http://revuln.com/files/ReVuln_Game_Engines_0days_tale.pdf
- http://en.wikipedia.org/wiki/Category:Lua-scripted_video_games
- http://www.garrysmod.com/updates/
- http://www.pcgamer.com/garrys-mod-cough-virus-is-cured-but-it-could-have-been-worse/
- http://www.garrysmod.com/2014/04/19/exploit-fix-released/
- http://www.valvetime.net/threads/gmod-has-a-lua-exploit-causing-mass-issues.244534/
- http://www.unknowncheats.me/forum/arma-2-scripting/70058-evil-scripts.html
- https://community.bistudio.com/wiki/
- https://gist.github.com/corsix/6575486
- http://www.fontspace.com/total-fontgeek-dtf-ltd/erbosdraco-nova-nbp
- http://newsaint.deviantart.com/art/shall-we-play-a-game-168941908 (image on the first slide is a modified version of this, released under CC BY-NC-SA 3.0 - http://creativecommons.org/licenses/by-nc-sa/3.0/)