

Anti-Virus Emulation Engine (AVE Engine)

Generic analysis of Win32 malware

© North Security Labs, 2011

<http://northsecuritylabs.com>

Current situation in malware world

- ▶ A lot of auto-generated obfuscated malware
 - ▶ Daily increase is about 10000 samples
- ▶ Custom packers and protectors
 - ▶ Cybercriminals earn money on them
- ▶ Growing complexity (anti-emulation and anti-debugging techniques)

Every AV needs emulator

No exceptions



Developing an emulator

- ▶ Time: 1...2 years depending on qualification of developers
- ▶ It is a big challenge to develop high-performance emulator
- ▶ Lot of pain (undocumented processor and OS features, API emulation, SEH, exceptions etc...)
- ▶ Open source solutions suffer from low performance and are not intended for use in anti-virus software



Solution

AVE Engine (Anti-Virus Emulation Engine) – a way to improve AV heuristic algorithms

- ▶ High performance due to dynamic binary translation technique (speed of unpacking is close to native)
- ▶ Low memory footprint
- ▶ Lot of supported known packers (UPX, ASPack, MEW, FSG, PECompact, NSPack, WinUpack and counting)
- ▶ Generic unpacking of custom packers used by malware authors
- ▶ Simple and flexible interface (setting intercepts of different kinds of events, setting emulation time limit)
- ▶ Ability to read emulated memory in event handlers for subsequent scanning
- ▶ Supported host platform are Windows x86 and Linux x86
- ▶ Supported emulated platform is Windows x86 (emulating execution of Win32 EXE and DLL files)
- ▶ Ready to use right now



Example of usage

```
h = ave_emul_init(); /* Creating instance of emulator. */
/* Setting some intercepts. */
ave_emul_setopt(h, AVE_EMUL_INTERCEPT_EVENT, AVE_EVENT_DATA_EXECUTED, 1);
ave_emul_setopt(h, AVE_EMUL_INTERCEPT_EVENT, AVE_EVENT_MODIFIED_CODE_EXECUTED, 1);
ave_emul_setopt(h, AVE_EMUL_INTERCEPT_EVENT, AVE_EVENT_UNHANDLED_API, 1);
/* Load data into emulator and start emulation. */
result = ave_emul_exec(h, buf, size, emul_callback, context);
...
ave_emul_shutdown(h);
...
static ave_result_t emul_callback(ave_handle_t h, int event_code,
                                void *data, size_t size, void *context)
{
    switch(event_code)
    {
        case AVE_EVENT_MODIFIED_CODE_EXECUTED: /* Read memory block and perform signature scan. */
            result = ave_mem_query(h, ...);
            ... break;
        ... /* Other events... */
    }
    ...
}
```



Licensing

- ▶ Licensing options
 - ▶ Binary per-project (LIB files and C header)
 - ▶ Source per-project (full source code)
- ▶ Update access and technical support
- ▶ Development of new features: on separate demand

Contact us for details: sales@northsecuritylabs.com

