
ParasiteEx

Disinfecting Parasitic Malware Platform-Independently

Thomas Barabosch, Adrian Dombek, Elmar Gerhards-Padilla
firstname.lastname@fkie.fraunhofer.de
Future Security 2015, Berlin



Cyber Analysis & Defense (CA&D)

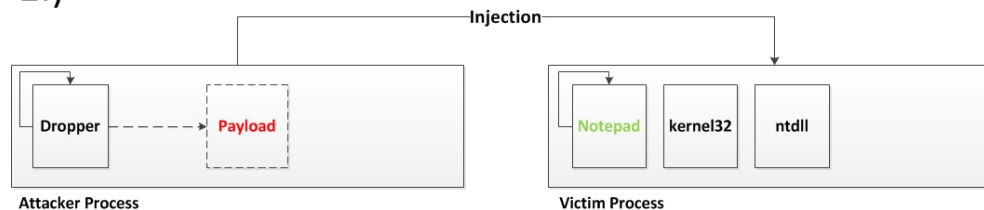


Source: <http://www.snoopwall.com/wp-content/uploads/2015/03/android-malware.jpg>

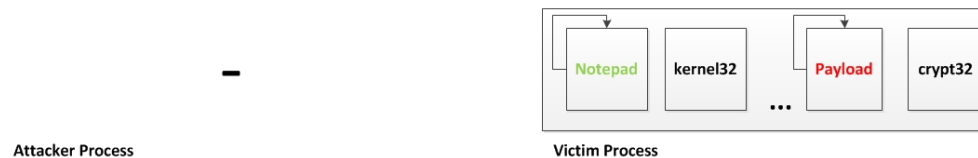
Host-Based Code Injection Attacks (HBCIAs)

- In a nutshell: run code in the context of another process
- HBCIAs consist of three steps [1]
 - Victim selection
 - Copying of code
 - Triggering of execution

1.)



2.)



Host-Based Code Injection Attacks (HBCIAs)

■ Benefits

- Covert operation
- Escalation of privileges
- Interception of critical information

■ Very popular with 2/3 of current Windows malware [1]

■ Several platforms are prone to this attack

Disinfection of HBCIA-infected Processes

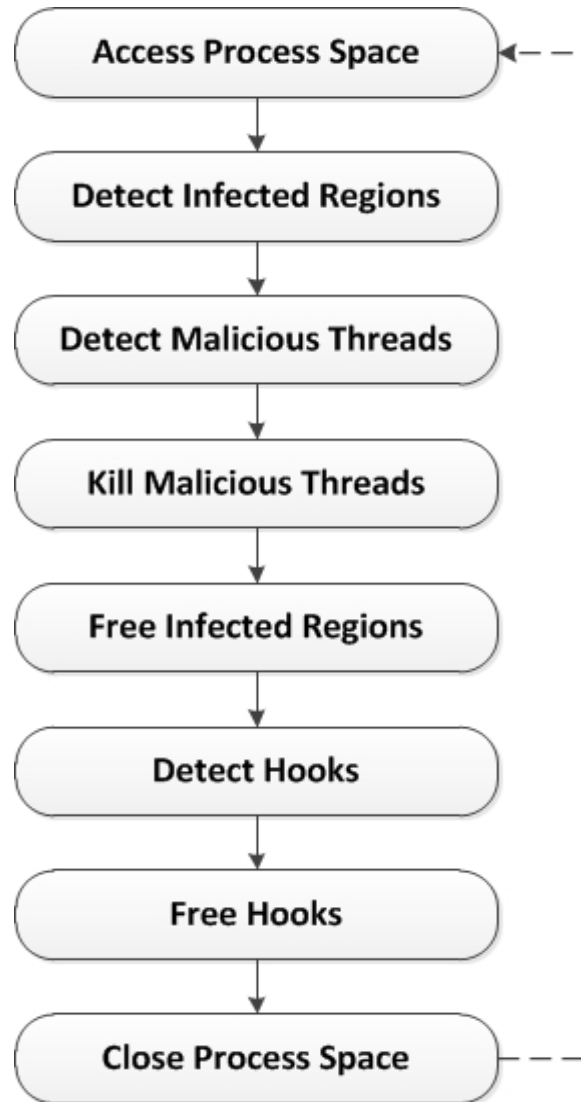
- Never trust an infected system
- However, in some cases a reboot is not possible (e.g. industrial plants)
- Systematic disinfection has been shunned
 - Works on detection and prevention (e.g. [2], [3], [4])
 - Disinfection of Conficker [5]

PARASITEEX

ParasiteEx

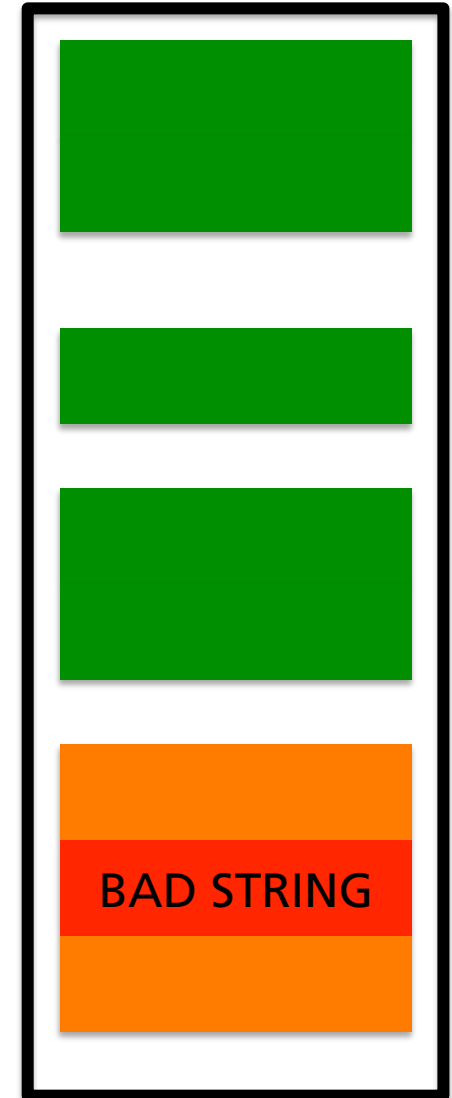
- Disinfects HBCIA-infected process spaces systematically
- Relies on common concepts found in almost all OSes
 - Processes (victim selection)
 - Memory regions (code copying)
 - Threads (execution triggering)
- Platform-independent
 - Prototype exists for Windows and Linux

ParasiteEx: Algorithm



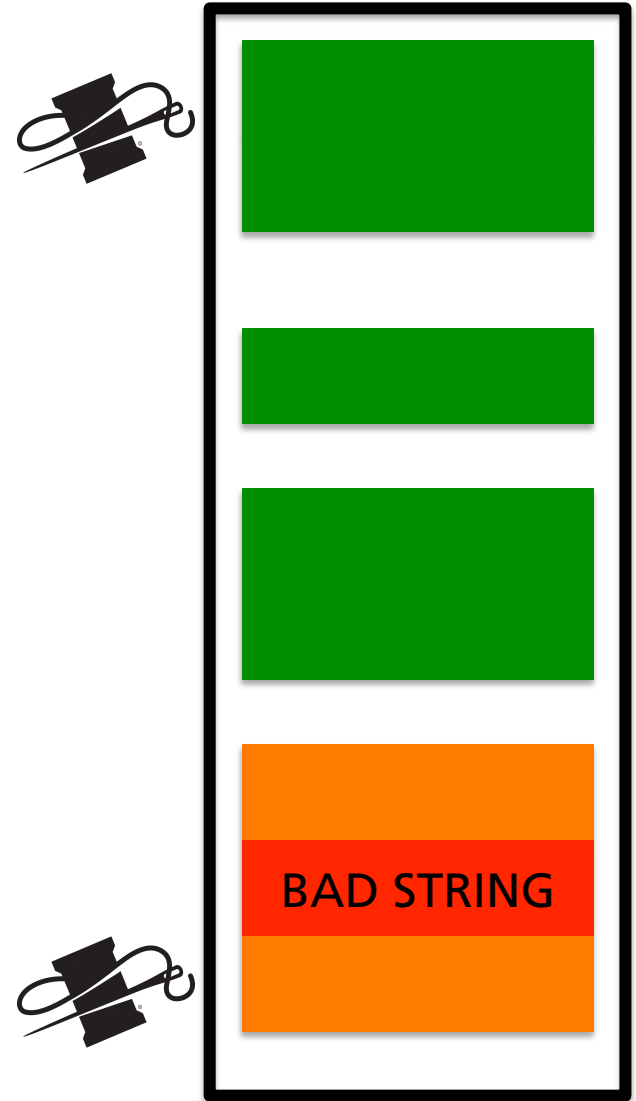
ParasiteEx: Infection Detection (1/2)

- Assumption: good set of signatures
- Scans all memory pages
 - If memory page matches signature -> whole region is assumed to be infected



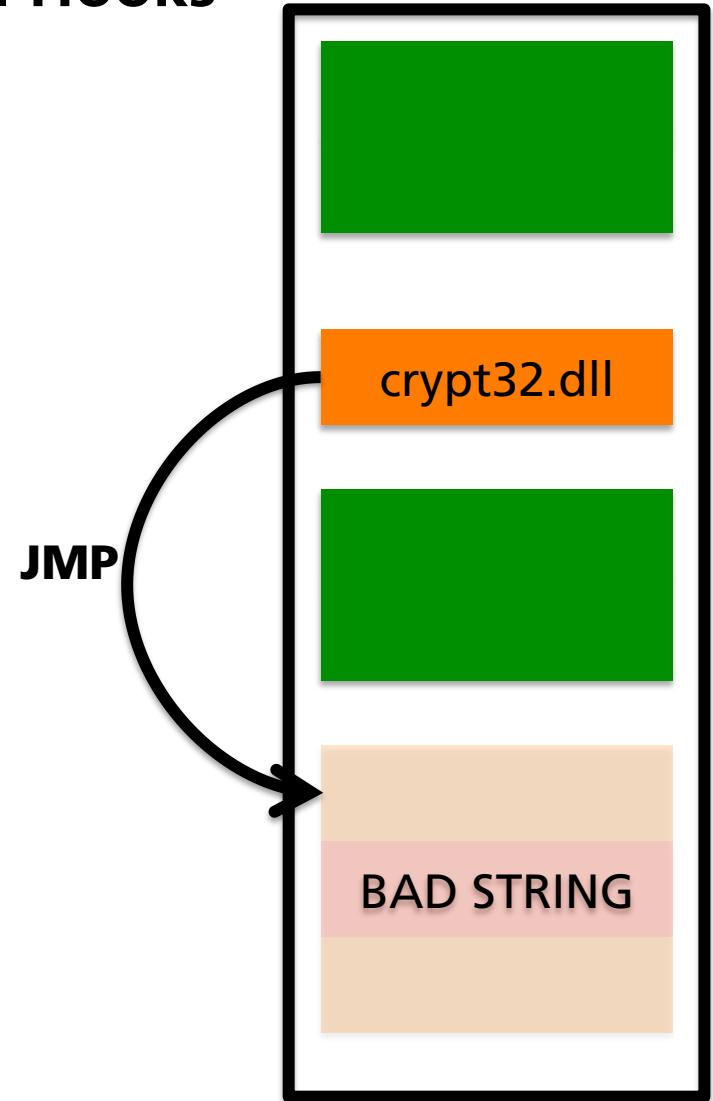
ParasiteEx: Infection Detection (2/2)

- Then malicious threads are determined
- A thread is malicious IFF it originates in infected region



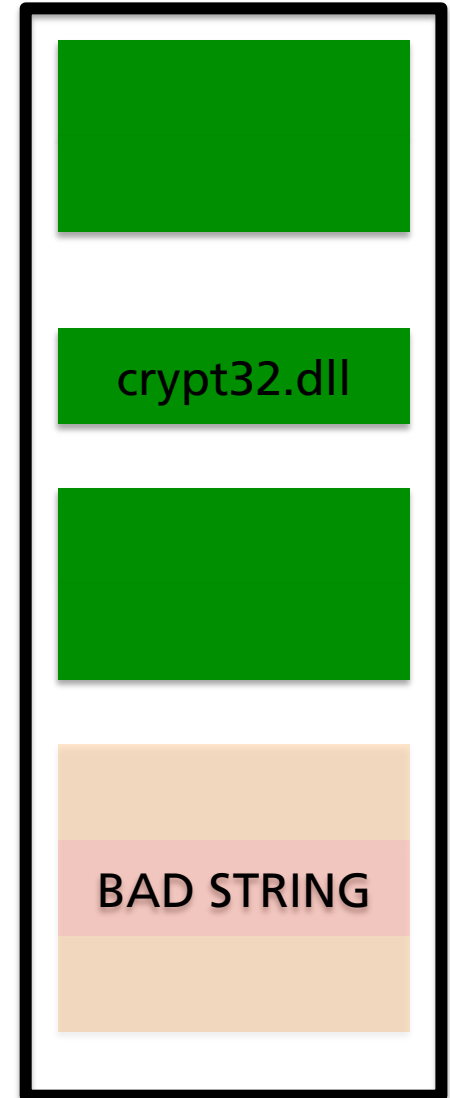
ParasiteEx: Detection and Removal Hooks

- Detects jumps from libraries to infected regions





ParasiteEx: Detection and Removal Hooks

- Detects jumps from libraries to infected regions
- Replaces affected libraries



EVALUATION

Data set

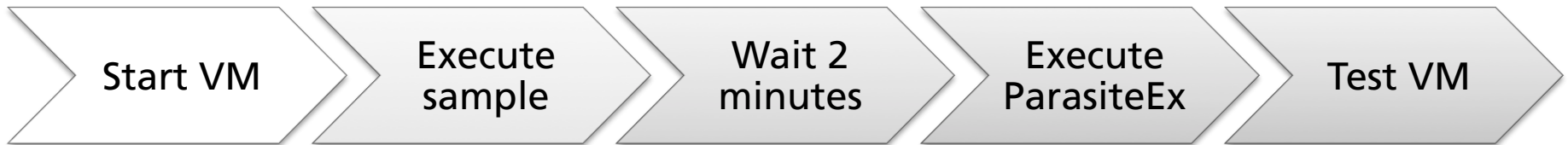
- Fifteen representatives of prevalent malware families
 - Bebloh, Sality, Vawtrack, ... 
 - Hanthie 
- 334 benign programs

Methodology

■ Environment

- Hardend VMs (Windows XP, Ubuntu Linux 13.10)
- No Internet connection

■ Preparation: Create signatures



Results

- No false positives
- Cleans successfully 11/14 Windows families
- Cleans successfully Hanthie on Linux
- Problems in three cases
 - Hooks
 - Detection of all malicious threads

FUTURE WORK & CONCLUSION

Future Work

- Improvement of hook detection & removal engine
 - Minimizing risk of system instability
- Dynamic Software Updating [6]
- Move ParasiteEx out of User Space



Conclusion

- ParasiteEx disinfects HBCIAs platform-independently
- Relies on concepts like threads or memory regions
- Prototype for Windows and Linux
- Prototype showed very promising results

References

- [1] T. Barabosch, S. Eschweiler, E. Gerhards-Padilla, Bee Master: Detecting Host-Based Code Injection Attacks, DIMVA 2014
- [2] T. Barabosch, E. Gerhards-Padilla, Host-Based Code Injection Attacks: A Popular Technique Used By Malware, MALCON 2014
- [3] G. S. Kc, A. D. Keromytis and V. Prevelakis, Countering code-injection attacks with instruction-set randomization, CCS 2003
- [4] A. Papadogiannakis, L. Loutsis, V. Papaefstathiou and S. Ionnidis, ASIST: architectural support for instruction set randomization, CCS 2013
- [5] F. Leder and T. Werner, Containing Conficker, 2009
- [6] C. Giuffrida, C. Iorgulescu, A. S. Tanenbaum, Mutable Checkpoint-Restart: Automating Live Update for Generic Server Programs, Middleware 2014