Botnet Analysis – II
(Dynamic Taint Analysis)

Amit Malik

www.SecurityXploded.com

Disclaimer

The Content, Demonstration, Source Code and Programs presented here is "AS IS" without any warranty or conditions of any kind. Also the views/ideas/knowledge expressed here are solely of the trainer's only and nothing to do with the company or the organization in which the trainer is currently working.

However in no circumstances neither the Trainer nor SecurityXploded is responsible for any damage or loss caused due to use or misuse of the information presented here.

Acknowledgement

- Special thanks to **Null** community for their extended support and co-operation.
- Special thanks to **ThoughtWorks** for the beautiful venue.
- Thanks to all the trainers who have devoted their precious time and countless hours to make it happen.

Advanced Malware Analysis Training

This presentation is part of our **Advanced Malware Analysis** Training program. Currently it is delivered only during our local meets for FREE of cost.

For complete details of this course, visit our Security Training page.

Who am I?

Amit Malik

- Member, SecurityXploded
- Security Researcher, McAfee Labs
- Reversing, Malware Analysis, Exploit Analysis/Development etc.
- E-mail: m.amit30@gmail.com

Content

- Recap
  - Botnets
  - Analysis techniques
- Automation and Our sessions
- Advanced Analysis and Detection Technologies
  - Execution flow graphs
  - Data flow graphs i.e dynamic taint analysis (DTA)
  - Exploit detection
  - Malware analysis and detection
- Finally, A joke (APT – Advanced Persistent Threat)

Recap

- In previous session we discussed,
  - Botnets
  - Rapid Reversing Techniques (RRT)
  - Waledac botnet analysis using RRT
- The RRTs we discussed earlier are the basic block of today's presentation
- We will cover automation in our upcoming sessions (details, next slide)

Automation and Our Sessions

- We will cover different aspects of automation in our upcoming sessions
  - Reversing Automation - Harsimran Walia
  - Sandbox or automated malware analysis systems – Monnapa
- Today's presentation is more on scientific solutions rather than normal automation stuff.

Advanced Analysis and Detection Technologies

- Security is a real complex problem at present.
- Threats are going more and more sophisticated.
- Traditional technologies are not enough to detect today's threats.
- So what we do now?
- Well, couple of technologies proposed earlier but DTA is the fascinating and powerful one, although used since 1989 ☺ (pearl programming language).
- Let's talk about the RRTs first and then DTA.

Execution Flow Graphs

- Using RRTs we generate graph and analyze the application.
- Can we use the same concept to solve another problem?

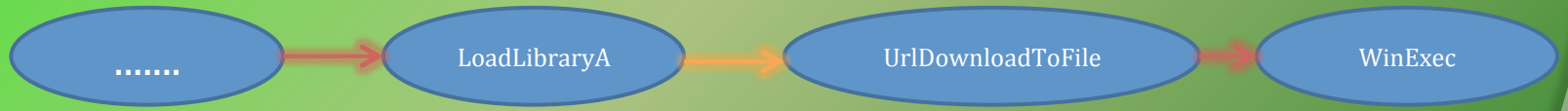
  "A sample is first executed on the virtual machine but didn't generated any network traffic, same sample again executed on the real system and this time generated the network traffic."

  *In minimum possible time identify the code segment which is responsible for detecting the VM or **deviating** the execution flow on virtual machine.*

Execution Flow Graphs

- Generate the execution flow graph on VM and real machine and then compare them.
- Example:

API Call Graph - VM



API Call Graph – Real System

- Instead of VM and real system let's call them state 1 and state 2.

Execution Flow Graphs Limitations

- Can’t use in detection, too coarse-grained approach.
- Good for analysis but not always.
- We need more fine-grained approach.
- “Data” is the most important point of the entire system.
- We need to track some specific data in order to claim some malicious behavior of any binary.

Dynamic Taint Analysis

- Track information or data flow inside binary during execution.
  - Information flow?
- What type of data?
  - Data from all untrusted sources, normally user input, file read, network read etc.
- Three main components
  - Taint source: user input, file read, network read etc.
  - Taint: data from taint sources (labeled data – memory start address and size, registers.)
  - Taint propagation: flow of tainted data in binary

Taint Propagation

- Data can be affected by two operations
  - Data movement operations
  - Arithmetic operations (Including all operations that are based on arithmetic operations like boolean etc.)
- IL (Intermediate Language)
- Taint Propagation
  - In data movement operation, destination will be tainted if and only if source is tainted.

Example: mov eax,tainted data

mov ebx,eax

here in 2nd instruction ebx is tainted because eax is tainted.

Taint propagation is transitive.

A => T(B), B => T(C) means A => T(C)

  - In arithmetic operation, result will be tainted if any byte or bit of the operands is tainted.
  - In some situations the above propagation methods may fail. eg: xor eax,eax, result should not be tainted in such cases.

Data (Taint) Flow Graph

- A graph can be generated based on how taint propagates.
- Resulting graph can be checked against the policies to detect the malicious behavior of binary.
- What policies?
  - Some rules that are either generated manually or learned by the machine to distinguish between normal data flow and malicious data flow.
  - Example: if a tainted variable is used in command execution on operating system then we have some serious problems.

DTA Applications

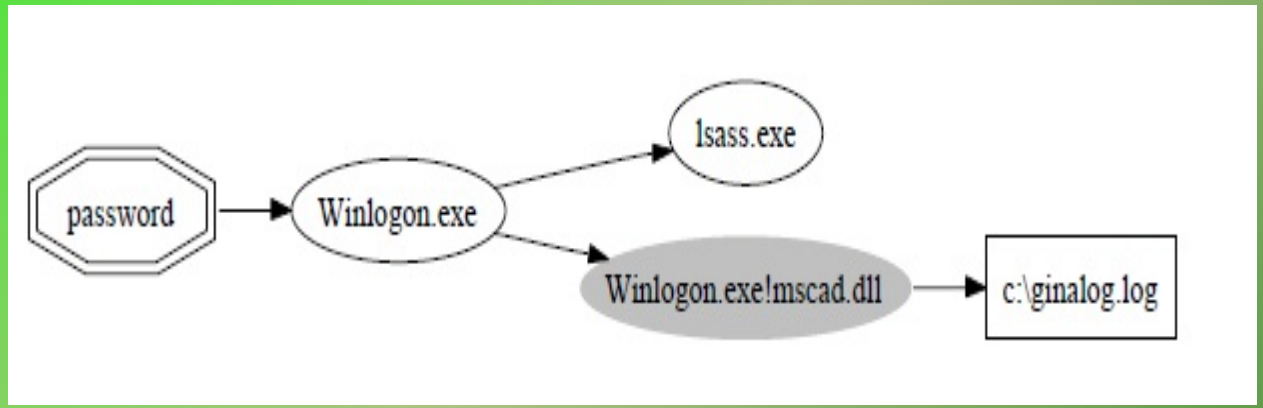
- Exploit Detection
  - If any time EIP points in user supplied data or in other words if EIP is in tainted memory/data range.
- Malware analysis and detection
  - Provides the answer to the question “how interested data is utilized by the application”
  - In-depth insight into the binary
  - Good analysis reports for forensic analysis, malware analysis
  - Detection can be done using some rules.

Key logger Detection using DTA

- Generate clean state (normal state) data flow graphs and use them as policies.
  - How user name and password data propagates in your browser?
  - How password data propagates during windows authentication, etc. ?
- In key logging
  - We will see the deviation in data propagation.
  - Clean state graphs works as a reference i.e data should be utilized by application according to the clean state graphs
  - In key logging the deviation of data flow trigger the suspicious behavior.

Cont..

- Graph from TEMU [see reference]

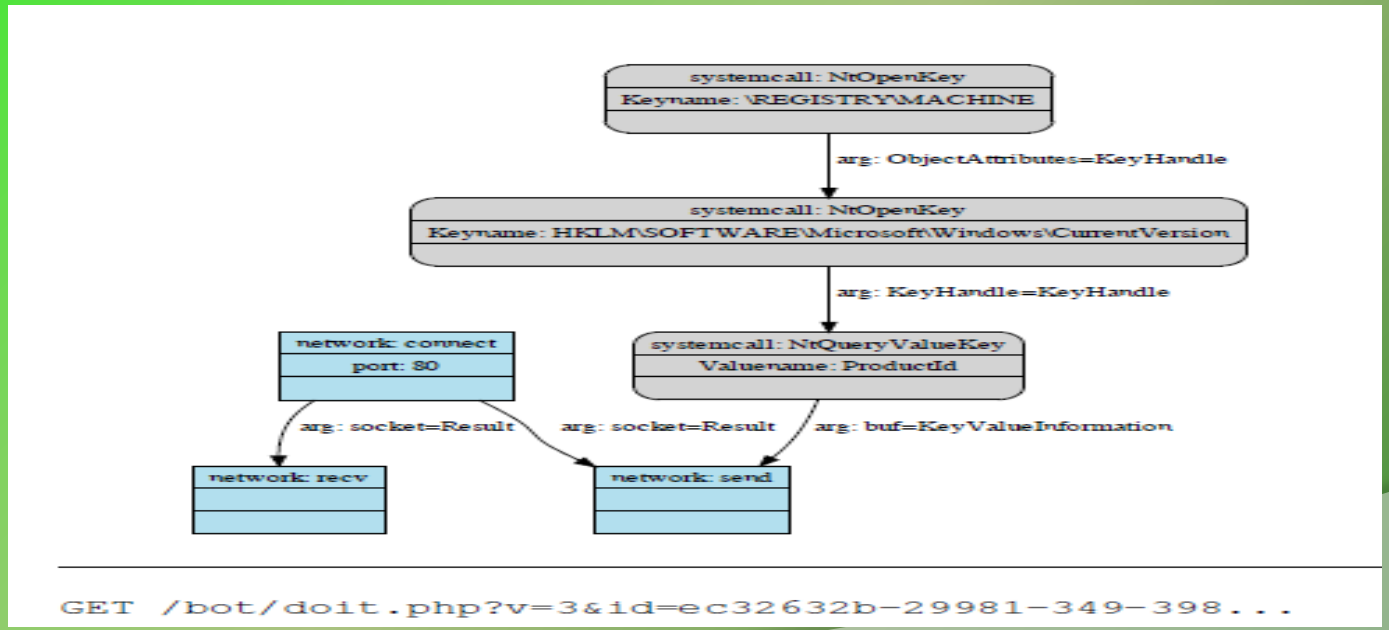


C&C Detection using DTA

- Initially discussed in JackStraws paper [see reference]
  - What type of data is sent by the application to the server?
  - What type of data is received by the application from the server?
  - *Correlate both type of data
  - See if combination* violates any normal behavior
- Example
  - Application read the machine ID, OS version from registry and send it to the server.
  - Server send some response after that application download a binary and executes the downloaded binary.
  - The above data propagation clearly denotes a malicious behavior.

Cont...

- Graph from Jackstraws [see reference]



Tools for Implementation

- We need to instrument two things
  - Data movement operations
  - Arithmetic Operations

  *Memory and registers
- Scope
  - Single process
  - Whole system
- Tools
  - DBI (Dynamic Binary Instrumentation) – PIN from intel
  - Qemu
  - Python (pydbg + pyEmu etc.)

DTA Limitations

- Can only explore single execution path
  - However, forward symbolic execution can be used in order to predict event based actions but still not very accurate.
- Too expensive for consumer products (slower execution etc.)
- Taint propagation methods can be evaded
- Complex implementation, usually combined with machine learning logics.

Few systems on DTA

- Dytan
- Valgrind (Plugins)
- TTAnalyze
- JackStraws
- BitBlaze (TaintQemu/TEMU)

APT

- Advanced Persistent threat
  - What do you think about “persistent” word here.
- Symptoms
  - Similar exe and dll names like system files
  - Similar registry key names like system registry keys
  - In some situations less noisy (low network traffic etc.)
  - Or may be event triggered (logic bombs)
- APT and you
  - For you APT is just a normal malware.

Reference

Complete Reference Guide for Advanced Malware Analysis Training

[Include links for all the Demos & Tools]

Thank You !

www.SecurityXploded.com