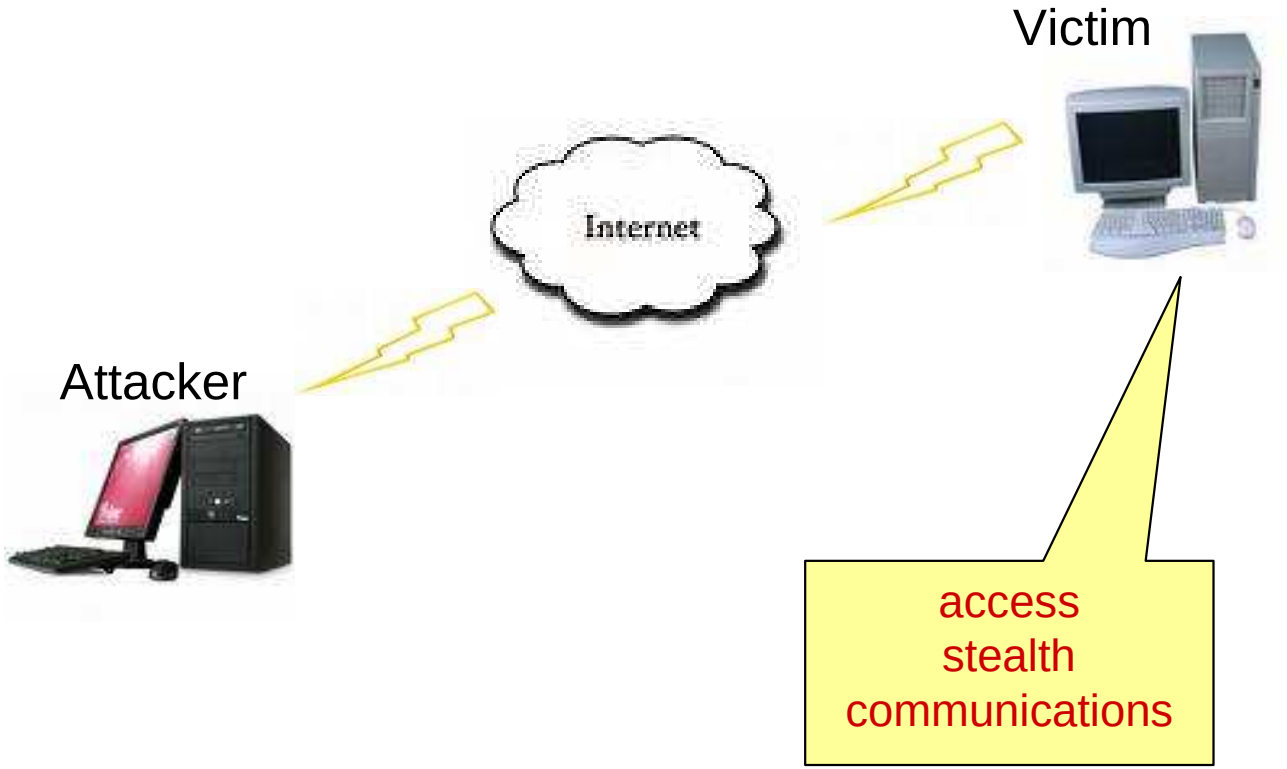
Inducing Observables to Detect Hidden Files and Processes on Computer Systems

Jim Jones

February 2007

What is a Rootkit?



Why Do I Care?

- Difficult to detect
- Maintain compromise
  - Remote access
  - Data/system
    - confidentiality
    - integrity
    - availability
- Mission critical systems, sensitive data

The Problem

- Current detection methods
  - Stealthy and non-stealthy
  - Known tool
  - Known method
- Novel methods
  - stealthy

Approach (1 of 2)

- Goal: Detect the presence of hidden files or processes, regardless of tool/method
- Consider:
  - Hidden process
  - Uses resources
    - CPU
    - Memory (kernel data structures)
  - System must (theoretically) respond differently if hidden processes (and files) exist

Approach (2 of 2)

- Approach:
  - Take actions on the system
  - Induce observables
  - Reason over observables
  - Infer presence of hidden file or process

Example: Process Creation (1 of 2)

- Hide backdoor process using FUto
  - Process name and PID not visible
  - Driver not visible
- Create multiple processes
- Collect PIDs

Example: Process Creation (2 of 2)

```
// CheckProcID.cpp
#include <iostream>
#include <Windows.h>
using namespace std;

int main ()
{
        DWORD pid;
        pid = GetCurrentProcessId();
        cout << pid << "\n";
        return (0);
}
```

```
C:\> FOR /L %A IN (1,1,1000) DO CheckProcID.exe >> out.txt
```

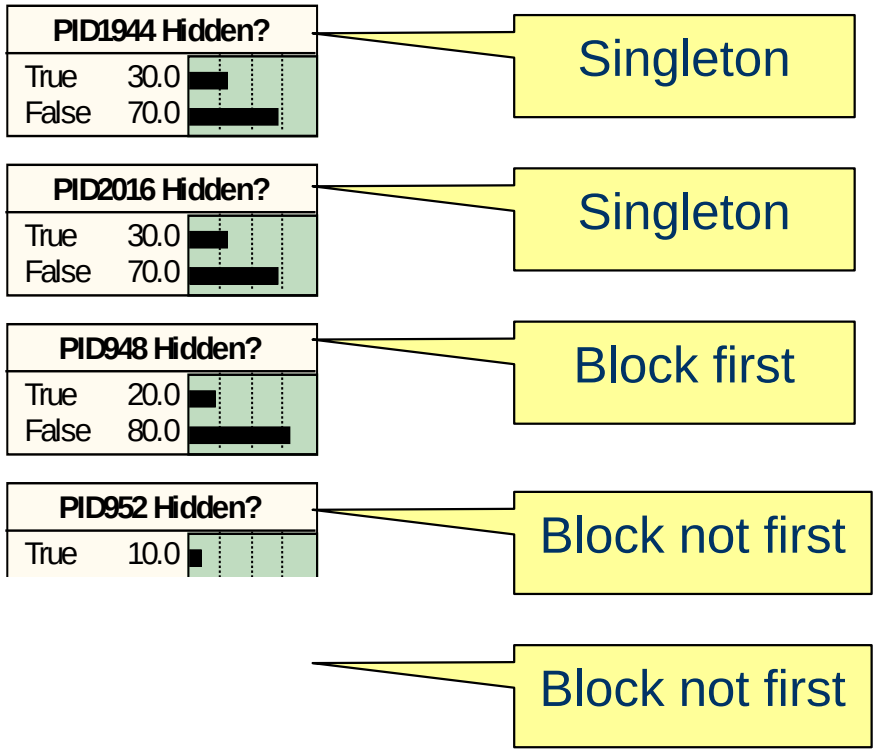
Process Creation Data (1 of 2)

- Multiple runs → 65535, 1,000,000, 10,000,000, 4,096, 675
- PID Findings:
  - Apparently random
  - In range 184-4092 (~2^12)
  - When sorted and uniqued
    - Always divisible by 4 (1028 possibilities)
    - 675 unique PIDs, for all runs over 4096
    - Multiple “gaps” in sequence, even at 10M iterations
    - Some gaps explained by running processes
    - Gaps often in blocks with known running process at start

Process Creation Data (2 of 2)

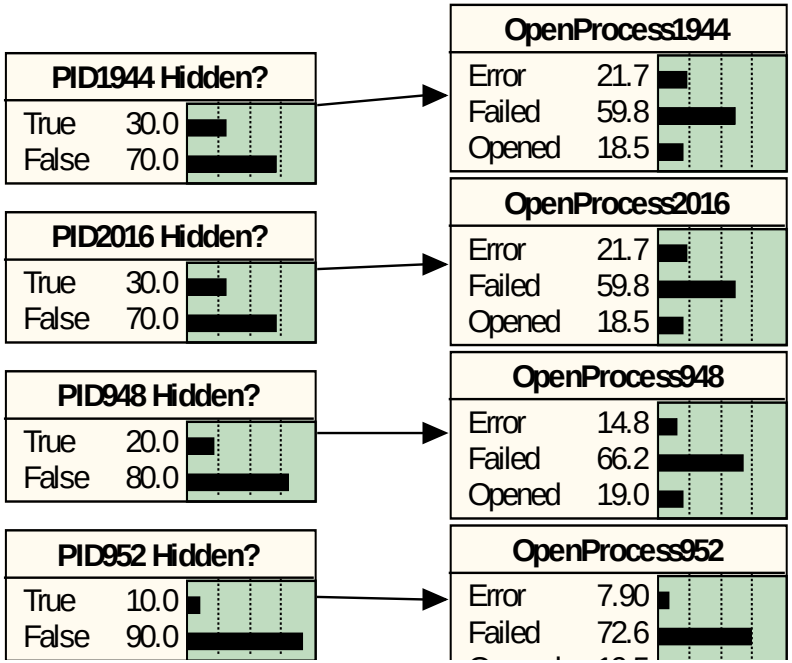
- Side Notes:
  - Analyze PIDs as time series?
  - Leave processes running?
- Conclusions:
  - Partial indicator of hidden process
  - Not conclusive
  - Need:
    - More evidence
    - Reason over combined evidence

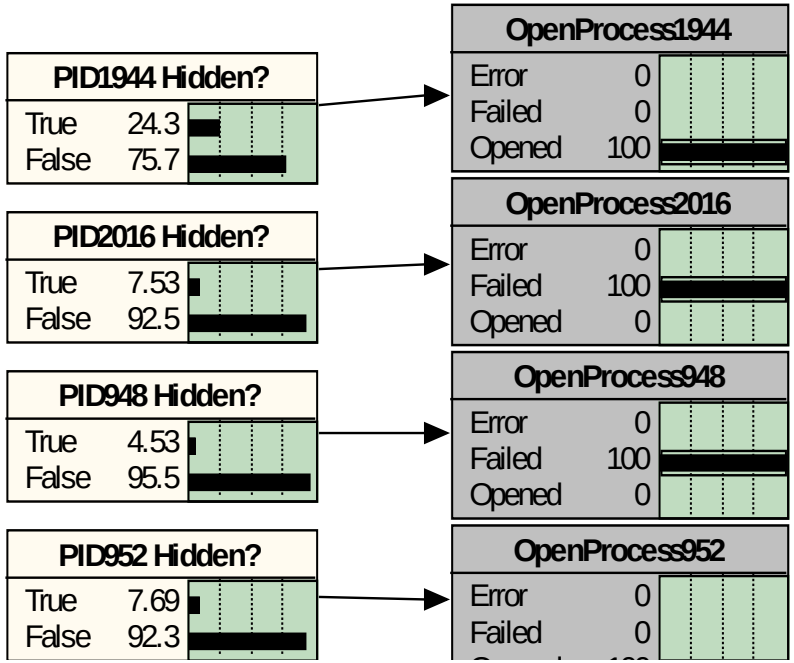
Bayesian Network Fragments



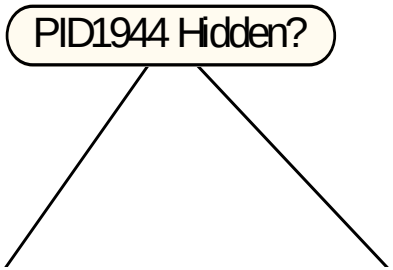
etc... (~300 total)

OpenProcess(PID)

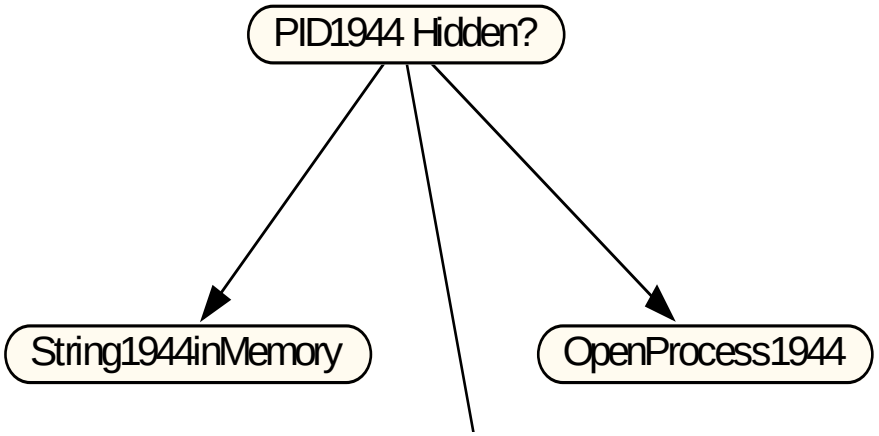




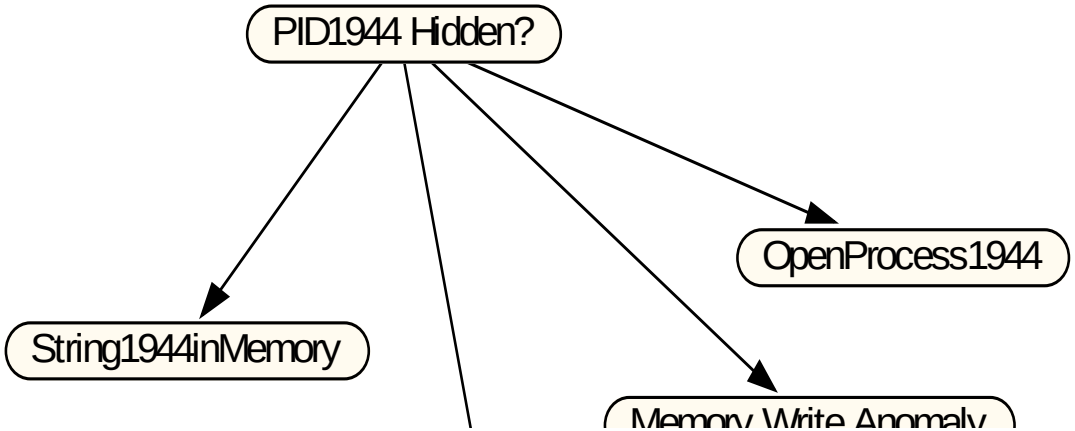
PID Strings in Memory



Memory Summation



Memory Write Error



On-going Work

- Additional process tests:
  - Thread count anomaly
  - Execution times
  - CPU usage (trigger hidden process)
- Similar test suite and model for hidden file

Limitations

- Threads
  - Top-down rootkits vs. bottom-up rootkits
- Moving targets
- Corrupt the tests

Thank You

GMU

SAIC

DARPA

HSARPA

IET, Inc.

HBGary, Inc.

Jim Jones

jonesj54@ferris.edu

(703) 629-9166