

# Stride Polymorphic SLED Detection Through Instruction Sequence Analysis

Presenter: John Kesapidis  
kesapid@ics.forth.gr

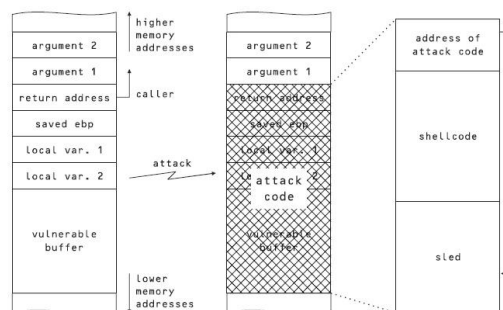
## Stride Polymorphic SLED Detection Through Instruction Sequence Analysis

1. Introduction
2. Classification of SLEDs
3. SLED Detection Mechanisms
4. Stride Algorithm
5. Evaluation
6. Performance
7. Conclusions
8. Questions

## Introduction

- Buffer Overflow Attacks
  - Aleph One (1996)
    - Tutorial for stack based buffer-overflow attack

## Introduction



*Figure 1.* Anatomy of a stack-based buffer overflow attack. By overflowing the buffer, the return address can be overwritten with a value pointing somewhere within the sled. The flow of control will be transferred to the start of the shellcode from any location in the sled.

## Introduction

- SLED
  - Fixes malicious code absolute addressing issue
  - Sequence of NOPs
- Detection of SLED can hint to a buffer overflow attack

## Classification of SLEDs

- Type 1
  - Simple sequence of NOPs

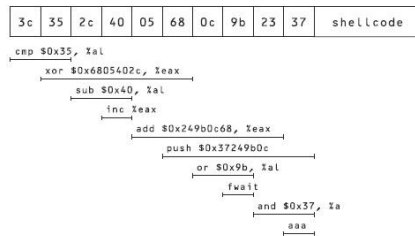
## Classification of SLEDs

- Type 2
  - One-byte NOP equivalent SLED
    - ADMmutate(55)
    - Metasploit Framework(+3)
    - 66 NOP equiv Intel IA-32

## Classification of SLEDs

- Type 3
  - Multi-byte NOP Equivalent SLED
  - Operands must be restricted to One-Byte NOP equivalent Instructions

## Classification of SLEDs



*Figure 2.* An example of a small sled, executable at every byte offset, which is constructed by interleaving one-byte and multi-byte NOP-equivalent instructions.

## Classification of SLEDs

- Type 4
  - 4-byte Aligned SLED
    - Assume word(4-byte) aligned stack
    - Pairs of 2-byte NOP equivalent Instructions

## Classification of SLEDs

- Type 5
  - Trampoline SLED
    - Jumps to Shell code directly
    - Assumes 4 byte alignment

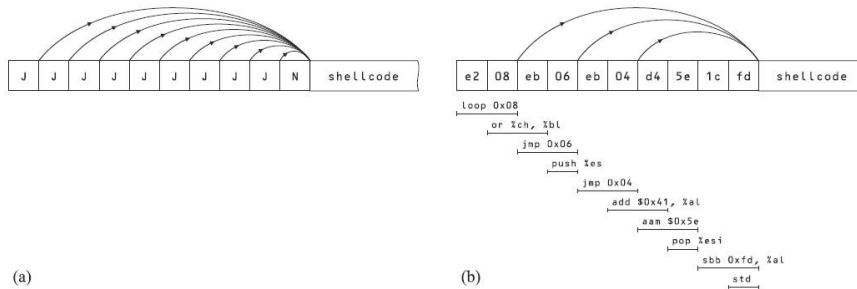


Figure 3. (a) The ideal trampoline-sled: flow of control is directed to the shellcode in a single step from any position in the sled. (b) An example of a small trampoline-sled that is executable at every byte offset. Control transfer instructions are placed at every second byte and their relative address operand is chosen so that it is a valid NOP-equivalent opcode.

## Classification of SLEDs

- Type 6
  - Obfuscated Trampoline SLED
    - Trampoline SLED Interleaved with NOPs

## Classification of SLEDs

- Type 7
  - Static Analysis Resistant SLEDs
    - Conditional Branches
    - Self-Modifying code

## SLED Detection Mechanisms

- NIDS Signatures
  - Rule Based(Scans for NOPs)
- Fnord
  - Scans for NOP-equivalent Instructions
- Abstract Payload Execution
  - Scans for Valid Instruction sequences

## SLED Detection Mechanisms

*Table 1. Comparative effectiveness of various sled detection schemes.*

Sled Type	Scheme			
	Snort	Fnord	APE	STRIDE
1. NOP instructions	Yes	Yes	Yes	Yes
2. One-byte NOP-equivalents	No	Yes	Yes	Yes
3. Multi-byte NOP-equivalents	No	No	Yes	Yes
4. Four-byte Aligned	No	No	Yes	Yes
5. Trampoline-sled	No	No	No	Yes
6. Obfuscated Trampoline-sled	No	No	No	Yes
7. Static Analysis Resistant	No	No	No	After extension*

## Stride Algorithm

```
stride(input, input_size, sled_length) {
    for (i=0; i < input_size-sled_length; i++) {
        if (find_sled(input+i, sled_length))
            return TRUE;
    }
    return FALSE;
}

find_sled(data, len) {
    for (j = 0; j < 4; j++)
        for (i = j; i < len; i+=4)
            if (!valid_sequence(data+i, len-i))
                return FALSE ;
    return TRUE;
}

is_valid_sequence(data, len) {
    /* decode "len" instructions in buffer "data" */
    res = decode(data, len);
    if (res == VALID_DECODE) return TRUE;
    if (res == ENDS_IN_JUMP) return TRUE;
    return FALSE;
}
```

Figure 4. Pseudo-code for STRIDE algorithm

## Evaluation

- 10,000 SLEDs (Metasploit Framework v2.2)
- Compare
  - Snort
  - Ford
  - APE
  - STRIDE

# Evaluation

Table 2. Detection rate of the various detection schemes for traces containing 10,000 different generated sleds of a single type.

Sled Type in Trace	Scheme			
	Snort	Fnord	APE	STRIDE
NOP instructions	100%	100%	100%	100%
One-byte NOP-equivalents	0%	55.4%	100%	100%
Multi-byte NOP-equivalents	0%	0%	100%	100%
Four-byte Aligned	0%	0%	100%	100%
Trampoline-sled	0%	0%	0%	100%
Obfuscated Trampoline-sled	0%	0%	Fig. 5	100%

# Evaluation

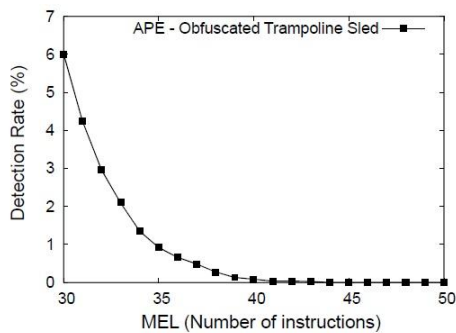


Figure 5. Detection rate for APE when applied to obfuscated sleds as a function of MEL.

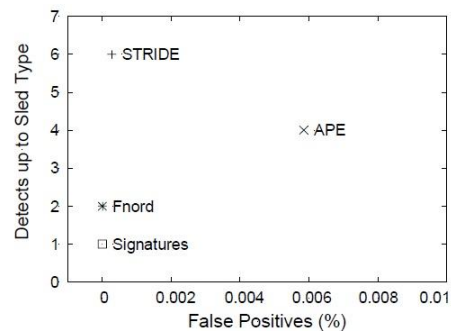


Figure 6. Comparative effectiveness of the various detection schemes. The results for APE are for a MEL value of 35 with 100 samples per kilobyte and for STRIDE for sled length 130 bytes.

## Evaluation

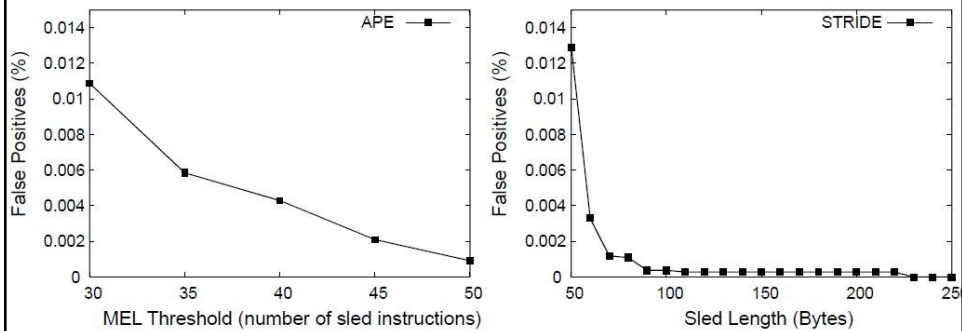


Figure 7. False positives rate for APE and STRIDE with varying parameters.

## Performance

- Stride is 5x faster than APE
  - APE follows both branch paths
  - Stride considers branches valid instructions but does not follow them

## Conclusions

- Stride detects more SLED types than previous proposals
- Stride achieves high detection rate with low false positive rate
- Stride is has relative low computation cost

Questions?